8215884

Halemane, Keshava Prasad

STUDIES IN THE OPTIMAL DESIGN OF FLEXIBLE CHEMICAL PLANTS

Carnegie-Mellon University PH.D. 1982

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages __√__
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __________ seem to be missing in numbering only as text follows.
13. Two pages numbered __________. Text follows.
14. Curling and wrinkled pages _____
15. Other ____

University
Microfilms
International

STUDIES IN THE OPTIMAL DESIGN

OF

FLEXIBLE CHEMICAL PLANTS

A Thesis Presented

by

Keshava Prasad Halemane

to

The Department of Chemical Engineering

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

Carnegie-Mellon University
Pittsburgh, Pennsylvania - 15213

February 24, 1982

Carnegie-Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE Studies in the Optimal Design of Flexible Chemical Plants

PRESENTED BY Keshava Prasad Halemane

ACCEPTED BY THE DEPARTMENT OF Chemical Engineering

MAJOR PROFESSOR 3/29/82 DATE

DEPARTMENT HEAD 3/29/82 DATE

APPROVED BY THE COLLEGE COUNCIL

DM Bernstein DEAN 4/2/82 DATE

ABSTRACT

The need to introduce *flexibility* in the optimal design of a chemical plant arises because of the fact that very often chemical plants have to operate under various different conditions. If these conditions are given as a discrete sequence of operating regimes, then the optimal design problem corresponds to a *deterministic multiperiod problem*. Also, there are situations when the values of some process parameters are known only approximately, or they actually vary or fluctuate during the operation of the plant. In such cases, the design strategy should be able to account for the *uncertainties* in the values of these *process parameters*. In both types of problems the main objective is to design a chemical plant that has the required flexibility to meet the specifications for the various conditions, while being optimal with respect to a selected economic index. This thesis addresses both these types of problems, giving the appropriate mathematical formulations and developing efficient solution procedures that can rigorously guarantee feasible steady state operation of the plant under the various conditions that may be encountered.

The *deterministic multiperiod design problem* is formulated as a nonlinear program which has a *block-diagonal* structure in the constraints. One of the main computational difficulties faced in the numerical solution of such problems is the large number of decision variables involved in the optimization. It is shown that the existing decomposition techniques cannot be applied effectively to circumvent this problem. A new decomposition scheme based on a *projection-restriction strategy* is developed, which exploits the

mathematical structure of the problem and the fact that many inequality constraints become active at the optimum solution. Successful implementation of this strategy requires an efficient method to find an initial feasible point, and the extension of current equation ordering algorithms for adding systematically inequality constraints that become active. Systematic procedures are proposed to handle both these aspects. The performance of this projection-restriction strategy is analyzed to show that significant gains in computational effort can be achieved with this technique. The results of the numerical example show a linear increase of computational time with the number of periods. Thus, the above strategy provides a real possibility for solving the large-scale nonlinear programs that arise in the design of flexible chemical plants.

In order to account for the uncertainties in the parameter values, a suitable strategy is to design a plant flexible enough, so that it can be guaranteed to satisfy the design specifications for a *given range of parameter values*, by suitable control in the operation of the plant. It is shown that a rigorous mathematical formulation of this design strategy corresponds to a *two-stage nonlinear infinite program*, wherein an optimization is to be performed on the set of design and control variables, such that the constraints are satisfied for every parameter value that belongs to a given bounded polyhedral region, while minimizing the expected value of the annual cost function. To guarantee feasibility of the design for every possible realization of the parameters, a logical constraint is incorporated in the problem. This *feasibility constraint* ensures that for every allowable value of the parameters (within the given range) that may be encountered during the operation, appropriate values of control variables can be chosen so as to satisfy the design specifications. In order to circumvent the problem of infinite dimensionality in the constraints, this feasibility constraint is shown to be equivalent to a *max-min-max* constraint, which also provides a deeper

understanding of the feasibility aspect of the design problem. It is shown that if the inequality constraint functions are convex, only the vertices of the polyhedral region in the parameter space need to be considered for design. Based on this feature, an algorithm is proposed which uses only a small subset of the vertices, and transforms the problem into an iterative multiperiod design problem. The proposed algorithm leads to an efficient solution procedure as shown by two example problems.

- KpH -

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my research advisor *Prof. Ignacio E. Grossmann*, for his valuable guidance throughout the course of this research work and in the preparation of this thesis. Working with him has been a unique experience for me.
I thank *Prof. Kenneth O. Kortanek* for devoting his time to the discussions we had on the infinite programming formulation. I appreciate the constructive criticism and useful suggestions of *Prof. Arthur W. Westerberg*. I also thank *Prof. Lorenz T. Biegler* and *Prof. Stephen W. Director* for their patience in reading through this thesis and for their comments.

The support, services and facilities provided by the Department of Chemical Engineering, Carnegie-Mellon University were essential means for my success in graduate studies and research. I also acknowledge the Design Research Center and the CIT Dean's office for the facilities provided. The financial support from the National Science Foundation under Grant CPE 79-26398 is greatly appreciated.

I cannot forget to mention my heart-felt gratitude towards my wife *Vijayalakshmi* for her love and understanding. I am grateful to my parents *Thirumaleshwari* and *Shama*, and my grandparents, for providing an atmosphere of encouragement in shaping me; and to my brothers and sisters for their affection.

I appreciate the help I received from my friends on various occasions. I thank Mr. Ross Swaney for reading this thesis and providing me with his comments.

- KpH -

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Chemical Plants are commonly designed for fixed nominal specifications such as capacity of the plant, type and quality of raw materials and products. Also, they are designed with a fixed set of predicted values for the parameters that specify the performance of the system, such as transfer coefficients or efficiencies and physical properties of the materials in the processing. However, chemical plants often operate under conditions quite different from those considered in the design. If a plant has to operate and meet the specifications at various levels of capacity, process different feeds, or produce several products, or alternatively when there is significant uncertainty in the parameter values, it is essential to take all these facts into account in the design. That is, a plant has to be designed *flexible* enough so as to meet the specifications even when subjected to various operating conditions.

In practice, empirical overdesign factors are widely used to size equipment, with the hope that these factors will compensate for all the effects of uncertainty in the design (Rudd and Watson, 1968). However, this is clearly not a very rational approach to the problem, since there is no quantitative

justification for the use of such factors. For instance, with empirical overdesign it is not clear what range of specifications the overdesigned plant can tolerate. Also, it is not likely that the economic performance of the overdesigned plant will be optimum, especially if the design of the plant has been optimized only for the nominal conditions.

In the context of the theory of chemical process design, the need for a rational method of designing flexible chemical plants stems from the fact that there are still substantial gaps between the designs that are obtained with currently available computer-aids and the designs that are actually implemented in practice. One of these gaps is precisely the question of introducing *flexibility* in the design of a plant. It must be realized that this is a very importar stage in the design procedure, since its main concern is to ensure that the plant will be able to meet economically the specifications for a *given range* of operating conditions. Clearly, it would be desirable to consider the dynamic performance characteristics of the chemical processing system in such a design procedure so as to ensure smooth operation of the plant. However, in order to provide a valid framework for such considerations, it is first of all necessary to *guarantee feasible steady state operation of the plant under the various conditions that may be encountered.* The term *flexibility* here indicates precisely this characteristic, namely that the existence of feasible steady state operation is ensured for every allowable situation that is specified, and it is this aspect that is studied in this thesis.

There are two classes of problems that can be considered in the design of flexible chemical plants. The first one is the *deterministic multiperiod problem* wherein the plant is designed to operate under various specified conditions in a sequence of time periods. Typical examples are refineries that handle various types of crudes, or pharmaceutical plants that produce several products. The second type of problem deals with the design of chemical plants where significant *uncertainty* is involved in the values of some of the

process parameters. Examples of this case arise when values of feed specifications, transfer coefficients, physical properties or cost data are not well established. It must be noted that in general a design problem can also be a combination of these two distinct types of problems.

1.1 REVIEW OF PREVIOUS RESEARCH WORK

Very little has been discussed in the literature about deterministic multiperiod problems. Loonkar and Robinson (1970), Sparrow et al. (1975), Oi et al. (1979), Suhami (1980) and Knopf et al. (1980) discuss design procedures that are applicable only to batch/semicontinuous processes. Grossmann and Sargent (1979) give a general formulation for designing multipurpose chemical plants which can also be applied to problems described by the deterministic multiperiod model. This involves the solution of a large nonlinear program (NLP), wherein the main computational difficulty is due to the large number of decision variables involved. Therefore, there is a clear need to develop efficient solution procedures for this problem. Chapters 2 and 3 present a decomposition scheme based on a projection-restriction strategy which is shown to be very efficient in solving this type of design problems.

There have been several approaches reported in the literature on the problem of design under uncertainty. They differ from each other in terms of problem formulation as well as solution strategies, since in principle the problem of design under uncertainty is not well-defined. Some authors consider the probability distribution of the parameters as either known or predictable and minimize the expected value of cost. Another approach consists in transforming the problem to a deterministic one and assuming that the parameters vary within bounded ranges of values that are specified by the designer. A brief overview of the main features of the different approaches is given below.

Kittrel and Watson (1966) assume that probability distribution functions of the parameters are available, and propose to select the decision variables in the design so as to minimize the expected value of cost. Wen and Chang (1968) define the 'relative sensitivity' of the cost as the fractional change in the cost function from its nominal value. In selecting the optimal design they minimize either the expected value or the maximal probable value of this relative sensitivity. Weisman and Holzman (1972) incorporate a penalty in the cost function involving the probability of violation for individual constraints, and perform an unconstrained minimization of the expected value of the cost. Although they made an attempt to minimize the probable violation of the constraints, their formulation does not ensure even a lower limit on the probability of failure of any given constraint, which can in fact be achieved by using the formulation suggested by Charnes and Cooper (1959) for chance constrained optimization. Lashmet and Szczepanski (1974) apply Monte Carlo simulation for determining overdesign factors for distillation columns. They perform a series of statistical experiments by choosing random values of the parameters (within the specified range), and in each case determine the number of stages in the column needed to meet the specifications. From these data the overdesign factor is determined as the additional number of stages corresponding to 90% cumulative distribution over that for nominal design. Freeman and Gaddy (1975) define 'dependability' as the fraction of 'time' that the process can meet the specifications, and use it as criterion for selecting the optimum design. They perform a stochastic simulation and determine the optimum values of the decision variables for different values of the parameters chosen randomly. The expected value of the cost corresponding to any given dependability is obtained from the results of such simulation, and the dependability level that minimizes the expected cost is chosen for the optimum design. The approach taken by Johns et al. (1976) is to select the optimum design by considering the effect of parameter uncertainties or time-varying demands on the expected value of cost computed over the whole life

of the plant. This procedure enables one to plan future expansions of existing plants and to analyze the economics of a new plant.

It is to be noted here that in the above approaches no distinction is made between the two types of decision variables in the problem of optimal process design. The *design variables*, representing for example the sizes of equipment, get their values assigned in the design stage, and remain unaltered during the operation of the plant. The *control variables* represent the variables of the plant that can be adjusted in the operation. For any given design, the optimal plant operation itself is to be considered as a means for meeting the specifications while minimizing the operating cost. It requires an appropriate choice for the values of control variables depending on the values of the *parameters* being realized. In a realistic strategy for optimal process design under uncertainty, the mathematical formulation of the design problem must incorporate this basic difference between design and control variables.

The following researchers have made such a distinction between design and control variables. Watanabe et al. (1973) apply the concept of statistical decision theory, by considering the problem of optimal process design as a two-person statistical game between Nature and the designer. They minimize a utility function which is a convex combination of the expected value and the maximum probable value of cost. That is, they follow a strategy which is intermediate between minimax strategy (maximal probable value of cost is minimized) and Bayes' strategy (expected value of cost is minimized). Avriel and Wilde (1969) discuss the different design strategies such as two-stage (here-and-now), wait-and-see and permanently-feasible programs. In a two-stage stochastic program, the designer selects values for the design variables (first stage), then observes the actual realization of the uncertain parameters, and accordingly chooses the appropriate values for the control variables (second stage). While selecting the values for design variables in the first stage, it is essential to ensure feasibility of the second stage sub-program,

namely that values of control variables can be chosen to satisfy the constraints. The objective is to minimize the expected value of cost while selecting a *feasible and optimal* design. Therefore, the *two-stage programming* formulation is certainly the most suitable representation for the problem of chemical process design under uncertainty. In the wait-and-see strategy, the designer waits for an observation of the uncertain parameters and then chooses the optimal values for both design and control variables. Here, each new value of the parameters results in a corresponding optimal design; or in other words, all decision variables are treated as control variables. In the permanently-feasible program, the designer selects (a single set of) values for both design and control variables which will be feasible for every possible realization of the uncertain parameters. Unlike the wait-and-see strategy, here the values of neither design nor control variables change with the variations in the values of the uncertain parameters. That is, in permanently-feasible program, all decision variables are treated as design variables. Avriel and Wilde (1969) suggest a procedure for obtaining the optimal design, which consists in bounding the objective function value that would be obtained at the solution of the two-stage program, by solving the wait-and-see program and permanently feasible program. However, they restrict their approach only to geometric programming formulations. Malik and Hughes (1980) apply a very similar approach for general process design problems, although there is no guarantee on the feasibility of the design. Also, the stochastic programming method they propose, based on Monte Carlo simulation, requires enormous computational effort.

Takamatsu et al. (1973) assume that the parameters vary within specified bounds, and minimize the deviation of the objective function from its value at the nominal solution, while satisfying the constraints linearized around their nominal values. They evaluate each constraint with a separate parameter value that would result in the worst violation of that constraint. This formulation

has a major limitation apart from the fact that the constraints are approximated by linearizations. Their approach is to seek a design that would meet the specifications with a single and common operating condition for all parameter values. This will (if at all it does) give a very conservative design, probably resulting in unnecessarily expensive investment. In other words, this approach does not recognize and take advantage of the fact that depending on the values of the parameters being realized, the operation of the plant can and need be manipulated in order to satisfy the specifications in the most economical way. Dittmar and Hartmann (1976) use a similar approach as Takamatsu et al. (1973), but suggest instead to use the same extreme value of the parameter for all the constraints. They determine the design margin for each of these extreme values and select the largest design margin so obtained. Nishida et al. (1974) propose a minimax strategy wherein the maximum value of the cost function, as obtained at the *worst parameter value* in a specified range, is minimized by selecting the appropriate design. They view the design strategy as a game, wherein the uncertainty in parameter values is considered to be resulting in the maximization of cost, whereas the objective of the designer is to minimize it. It is important to note here that the design they come up with, actually corresponds to the optimum solution for a particular (namely, the *economically worst*) value of the parameters, and therefore the design cannot be claimed to be optimal in an overall sense when the parameters do take on different values. Also, the feasibility of this design for other values of the parameters cannot be guaranteed, since this aspect has not been explicitly considered in the problem formulation. Friedman and Reklaitis (1975a,b) deal with linear programming formulations having uncertainties in the coefficients. They show that this formulation can be applied to problems like planning future operation policies for large interacting systems, production scheduling, resource allocation and determining optimum blending schemes. They incorporate required flexibility in their system by allowing for possible future additive 'corrections' on the current decisions, and

optimize the system by applying an appropriate cost-for-correction in the objective function. It is interesting to see that they were able to identify the need for different corrections for different outcomes of the uncertain coefficients, in order to make the problem feasible, and devise a procedure to achieve this in their computations. One obvious drawback with their approach is that it is applicable only to linear systems. A second limitation with their approach is that it is not suited for the problem of optimal design of flexible chemical plants, because in this case no additive corrections can be applied on the design variables. Instead, it is only the control variables that can be manipulated, so as to meet the specifications in spite of the variations in the values of the uncertain parameters. Kilikas and Hutchison (1980) use a linear process model wherein the coefficients are considered to be varying within specified bounds. However, their approach is not applicable for design purposes, because they tend to select the optimum value of their decision variables so as to satisfy their constraints for *at least one* set of parameter values. This objective is quite different from what is expected of a design strategy. In fact, they solve a steady state linearized simulation problem, with no mention about design variables, and therefore the optimization that they refer to seems to be only a suitable criterion for solution of the particular system of equations they present.

With any of the above approaches, one is always faced with the question as to whether the designed plant can in fact be guaranteed to operate and satisfy the specifications for the entire range of parameter values involved. This question, along with the fact that the problem of optimal process design under uncertainty is not well-defined, requires a systematic procedure in formulating as well as solving such design problems. First of all, it is to be noted here that *while minimizing the cost, the main concern of a design engineer must be to ensure feasible steady state operation of the plant for every value of the parameters within specified bounds.* Grossmann and Sargent (1978)

propose a formulation that tries to incorporate this objective. They approximate the expected value of the cost by a weighted average of a finite number of terms, assuming discrete probabilities for a finite set of parameter values. They select the optimum design by minimizing this expected cost subject to maximizing each of the individual inequality constraints with respect to the parameters. In their solution procedure a small set of extreme values of parameters is selected by analyzing the signs of the gradients of each of the individual inequality constraints, and the optimization is performed for this set of parameter values, in the form of a multiperiod design problem. However, as it is shown in Chapter 5, their approach cannot always guarantee the *feasibility requirements* mentioned above.

Among the literature on optimal design under uncertainty in other engineering fields, Kwak and Haug (1976) formulate the problem of 'parametric optimal design' wherein the current decision is chosen so as to maintain future feasibility even in the presence of uncertainties (with no possible corrections on the decision that has already been taken). This leads to a nonlinear program wherein the constraints correspond to individual maximization (in the space of the uncertain parameters) of each of the constraint functions. They show the application of this approach in designing structural (civil engineering) elements. However, the major difference here is that once the design is selected, no further adjustments are possible - unlike the case of chemical process systems. In fact, it is interesting to see that although the freedom in future decisions enables a given design to remain feasible by incorporating the appropriate adjustments, it requires more complex mathematical representation, resulting in computational difficulties. In structural design it is usually the external load that may be uncertain, and there seems to be no adjustable variables available, the values of which can be chosen along with changes in load. The problem of design centering, tolerancing and tuning of electrical/electronic circuits has been studied extensively, in electrical

engineering literature. One objective there is to maximize the yield by selecting the appropriate design center and assigning the right tolerance, so that the effect of any uncertainties in the 'design parameters' caused during manufacture can be compensated by applying the required tuning. See for example, Bandler (1974), Bandler et al. (1975), Director and Hachtel (1977), Polak and Sangiovanni-Vincentelli (1979). However, there the uncertainty occurs in the manufacturing, and is associated with the 'design parameters' that define the circuits, with a difference that these can be tuned by a later finishing process which will try to compensate directly, for the effect of the uncertainties. It is to be noted here that the problem of optimal design of chemical plants with uncertain 'process parameters' is different in its nature from the above two types of design problems. This is because the uncertainties for this case correspond to the process parameters once a given design has been chosen, unlike the design of structural elements or integrated circuits. Hence, the methods proposed for the above types of design problems cannot be applied readily to the design of chemical plants.

1.2 OBJECTIVES OF THIS RESEARCH WORK

This thesis addresses both the *deterministic multiperiod* problem and the problem of *design under uncertainty.* In Chapter 2 an efficient decomposition scheme based on a *projection-restriction strategy* is presented for solving the large nonlinear programming (NLP) problem that arises in formulating the deterministic multiperiod model. This decomposition strategy effectively exploits the *block-diagonal structure* in the constraints of this NLP, and also the fact that many inequality constraints become active at the solution of a design problem. Efficient methods have also been proposed for finding an initial feasible point for the design problem, and for detection of redundancy and selection of decision variables for the system of equations in the resulting optimization problem. Computational results obtained by the application of the proposed decomposition strategy are discussed in Chapter 3.

A rigorous mathematical formulation for the problem of design under uncertainty and its properties are presented in Chapter 4. It is shown that this design problem is represented by a *two-stage nonlinear infinite program*, wherein optimization is to be performed on the set of design and control variables so as to satisfy the constraints for all possible values of the parameters (within a specified bounded polyhedral region), while minimizing the expected value of the annual cost function. The feasibility of the design is expressed by a logical constraint, which ensures that for every allowable value of the uncertain parameters that may be encountered during operation, appropriate values for control variables can be chosen to manipulate the operation so as to satisfy the constraints, while minimizing the operating cost. The main computational difficulty with this formulation arises because of the infinite number of constraints and infinite number of variables involved. By selecting a finite number of parameter values within the given bounds, and approximating the expected value of cost function by a weighted average, the number of variables can be reduced to be finite. To overcome the problem of an infinite number of constraints, an equivalence is established for the *feasibility constraint* which leads to a *max-min-max constraint*. This max-min-max constraint provides information regarding the extent of (in)feasibility of a given design. For the case of a convex feasible region, it is shown that the solution to the max-min-max problem lies at a vertex (corner point) of the polyhedron in the parameter space. Based on these features, a solution algorithm is presented in Chapter 5, which incorporates the projection-restriction strategy (as in the multiperiod design problem) in an *efficient iterative scheme*. The application of the proposed solution algorithm and the computational results are also discussed in Chapter 5.

CHAPTER 2

THE DETERMINISTIC MULTI-PERIOD DESIGN PROBLEM

There are many situations when a chemical plant has to operate under more than a single set of operating conditions. For example a plant may have to operate at various levels of capacities, process different types of feeds, or produce several products. This can be achieved only if the plant is designed with the required flexibility, so as to meet the specifications even when operating under various different conditions. When these conditions are either known or predictable, the design problem can be formulated as a deterministic multiperiod problem. Since in such a formulation the specifications for each of the various operating conditions are explicitly incorporated, the resulting design will naturally satisfy the feasibility requirements. The design problem is in fact formulated as a nonlinear programming (NLP) problem wherein the annual cost is minimized, subject to the constraints that represent the design specifications. This has been the general approach proposed by Grossmann and Sargent (1979), and is presented below.

2.1 MATHEMATICAL FORMULATION

In the deterministic multiperiod model it is assumed that the plant is subjected to piecewise constant operating conditions in N successive time periods. Dynamics are neglected, as it is considered that the length of the transients is much smaller than the time periods for the successive steady states. The variables in this problem are partitioned into three categories. The vector d of *design variables* is associated with the sizing of the units. These variables remain fixed once the design is implemented, and do not vary with the changes in the operation of the plant. The vector z^i denotes the *control variables*, the values of which can be suitably chosen in each period i so as to meet the specifications and also minimize the operating cost. It should be noted that the vector z^i corresponds to the existing degrees of freedom in the operation of the plant. Finally, the vector x^i corresponds to the *state variables* in the operating period i (i=1,2,....N). Thus, the design problem leads to the nonlinear program,

$$\min_{d,\, z^1, z^2, \dots z^N} \quad C = C^o(d) + \sum_{i=1}^{N} C^i(d, z^i, x^i)$$

$$\text{s.t.} \quad \left.\begin{aligned} h^i(d, z^i, x^i) &= 0 \\ g^i(d, z^i, x^i) &\leq 0 \end{aligned}\right\} \quad i = 1,2,\dots N \tag{2.1}$$

$$f(d) \leq 0$$

where d, z^i, i=1,2,....N, are the decision variables in this problem, as the state variables x^i, i=1,2,....N, can be determined from the equality constraints which represent the steady state performance of the process system. Note that in this formulation the order in which the periods are considered can be arbitrary, since the operation in each period is assumed to be independent of its relative position in the sequence. However, any specifications involving all the periods can in general be represented by the last constraint, in which case f would be a function of d, z^i, x^i, i=1,2,....N.

2.2 COMPUTATIONAL ASPECTS

For large industrial problems the computational requirements for solving the NLP in (2.1) can become rather expensive. The reason for this is that the number of control variables z^i increases with the number of periods N, so that the number of decision variables in the NLP may become too large for the problem to be solved efficiently by the current algorithms. Since the NLP approach for designing flexible plants has proved to be very effective in small problems (Grossmann and Sargent, 1979), and it also provides a rational basis for overdesign, there is a very high incentive for deriving an efficient method for solving problem (2.1). This requires that its mathematical structure be fully exploited.

In order to take advantage of the sparsity of the constraints, the state variables x^i can be eliminated from the system of equations so as to reduce the size of the problem. This can be achieved, for instance, if the system of equations is ordered so as to provide a sequence of calculation where the number of torn variables is minimized (see e.g. Christensen, 1970). In this scheme, at each iteration of the optimization the ordered system of equations is solved. It must be pointed out, however, that by eliminating the equations and state variables the nonlinear program in (2.1) still has to handle the large number of decision variables given by d, z^i, i=1,2,....N. Therefore, it is necessary to exploit additionally another property of (2.1) for deriving an efficient method of solution.

Problem (2.1) has the interesting feature that it is an NLP with *block-diagonal* structure in the constraints, as shown in Figure 2.1. Since the cost function is separable in the N periods this implies that if the vector d is fixed, then the problem decomposes in N uncoupled subproblems, each having as decision variables the vectors z^i, for i=1,2,....N. This would suggest that it should be possible to derive a suitable decomposition scheme which need not

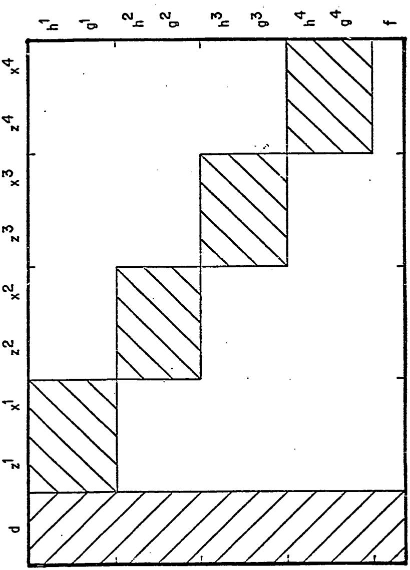

Figure 2.1: Block-diagonal Structure in the Constraints of Problem (2.1)

handle simultaneously all the decision variables. Ideally, such a decomposition scheme should lessen the storage requirements, and more importantly it should reduce substantially the computational time and also enhance the reliability of the method for obtaining the solution.

2.3 DECOMPOSITION STRATEGIES

The two basic decomposition strategies that could be used for solving problem (2.1) are the *feasible* and the *infeasible* decomposition schemes.

The feasible decomposition technique (Rosen and Ornea, 1963; Umeda, Shindo and Tazaki, 1972) consists of the following steps:

Step 1 - Find a feasible point d, z^i, i=1,2,....N, for problem (2.1).

Step 2 - By keeping the vector d fixed, solve the N subproblems (that is, for i=1,2,....N):

$$\begin{aligned} &\underset{z^i}{\text{minimize}} && C^i(d,z^i,x^i) \\ &\text{s.t.} && h^i(d,z^i,x^i) = 0 \\ & && g^i(d,z^i,x^i) \leq 0 \end{aligned} \tag{2.2}$$

Step 3 - Keeping the vectors z^i, i=1,2,....N fixed, solve the problem:

$$\begin{aligned} &\underset{d}{\text{minimize}} && C = C^o(d) + \sum_{i=1}^{N} C^i(d,z^i,x^i) \\ &\text{s.t.} && \left.\begin{aligned} h^i(d,z^i,x^i) &= 0 \\ g^i(d,z^i,x^i) &\leq 0 \end{aligned}\right\} \; i=1,2,....N \\ & && f(d) \leq 0 \end{aligned} \tag{2.3}$$

Step 4 - If convergence is not achieved, return to step 2.

The advantage with this technique is that the original problem is replaced by a sequence of subproblems with a smaller number of decision variables. However, convergence to the solution can become extremely slow, particularly in the neighborhood of the solution (see Grigoriadis, 1971), since in fact this decomposition technique is equivalent to an alternating search in orthogonal directions in the space $(d, (z^1, z^2, \ldots z^N))$

In the infeasible decomposition technique (Brosilow and Lasdon, 1965; Lasdon, 1970; Stephanopoulos and Westerberg, 1975b) it is first necessary to reformulate problem (2.1) as:

$$\begin{aligned}
&\underset{\substack{\hat{d}, d^1, d^2, \ldots d^N \\ z^1, z^2, \ldots z^N}}{\text{minimize}} \; C = C^o(\hat{d}) + \sum_{i=1}^{N} C^i(d^i, z^i, x^i) \\
&\text{s.t.} \quad \left.\begin{aligned} h^i(d^i, z^i, x^i) &= 0 \\ g^i(d^i, z^i, x^i) &\leq 0 \\ f^i(d^i) &\leq 0 \\ \hat{d} &= d^i \end{aligned}\right\} \; i = 1, 2, \ldots N \\
&\qquad\quad f(\hat{d}) \leq 0
\end{aligned} \tag{2.4}$$

Since the lagrangian of this problem is given by:

$$\begin{aligned}
L = C^o(\hat{d}) &+ \sum_{i=1}^{N} C^i(d^i, z^i, x^i) \\
&+ \sum_{i=1}^{N} \left[(\lambda^i)^T h^i + (\mu^i)^T g^i + (\nu^i)^T f^i + (\pi^i)^T (\hat{d} - d^i) \right] \\
&+ \rho^T f
\end{aligned} \tag{2.5}$$

where λ^i, μ^i, ν^i, π^i, ρ are the Kuhn-Tucker multipliers, problem (2.4) can be decomposed into the following N+1 subproblems:

$$\begin{aligned} &\underset{d^i, z^i}{\text{minimize}} \quad C^i(d^i,z^i,x^i) - (\pi^i)^T d^i \\ &\text{s.t.} \qquad h^i(d^i,z^i,x^i) = 0 \\ &\qquad\qquad g^i(d^i,z^i,x^i) \leq 0 \\ &\qquad\qquad f^i(d^i) \leq 0 \end{aligned}$$

for i=1,2,....N, and (2.6)

$$\begin{aligned} &\underset{\hat{d}}{\text{minimize}} \quad C^o(\hat{d}) + \sum_{i=1}^{N} (\pi^i)^T \hat{d} \\ &\text{s.t.} \qquad f(\hat{d}) \leq 0 \end{aligned}$$

The infeasible decomposition strategy then consists of the following steps:

Step 1 - Guess the multipliers π^i, i=1,2,....N.

Step 2 - Solve the N+1 subproblems (2.6)

Step 3 - If the constraints $d = \hat{d}^i$, i=1,2,....N, are not satisfied, adjust the multipliers π^i by solving the dual problem:

$$\underset{\pi^1,\pi^2,\ldots\pi^N}{\text{minimize}} \quad C^o(\hat{d}) + \sum_{i=1}^{N} \left[C^i(d^i,z^i,x^i) + (\pi^i)^T (\hat{d} - d^i)\right] \tag{2.7}$$

and then return to step 2.

Note that in this decomposition technique it is not necessary to start with a feasible point for problem (2.1) as with the previous strategy. However, there are basically two difficulties when using this technique. The

first one is that the method may not converge to the solution due to the nonconvexities that are present in design problems which gives rise to dual gaps. This difficulty can be overcome with the method proposed by Stephanopoulos and Westerberg (1975a), but with the disadvantage that it requires a significant amount of computational effort. A further disadvantage with the infeasible decomposition scheme is that a feasible solution is obtained only at the exact solution of the dual problem. Considering that one is dealing with nonlinear problems, this can become a significant drawback in practice.

With the two decomposition schemes that have been presented above, it is unclear whether problem (2.1) can be solved more efficiently than when the problem is tackled with all decision variables simultaneously. It is for this reason that an alternative decomposition strategy must be considered.

2.4 THE PROJECTION-RESTRICTION STRATEGY

Grigoriadis (1971) and Ritter (1973) have suggested a decomposition technique for mathematical programming problems with block-diagonal structure in the constraints, for the case when the objective function is convex and the constraints are *linear*. As per the classification given by Geoffrion (1970) this strategy belongs to the class of *Projection-Restriction Strategies.* The basic ideas of this strategy are described in the following steps:

Step 1 - Find a feasible point d, z^i, x^i, $i=1,2,\ldots N$, for the problem (2.1)

Step 2 - (Projection)

Fixing the values of the design variables d,
solve the N subproblems in (2.2)

Step 3 - (Restriction)

(a) For each subproblem i convert the n^i_A inequality constraints g^i_A that are active in step 2 into equalities, and define

$$h_R^i = \begin{bmatrix} h^i \\ g_A^i \end{bmatrix}, \quad g_R^i = g_I^i, \quad i=1,2,\ldots N \tag{2.8}$$

where h_R^i, g_R^i are the redefined sets of equality and inequality constraints and g_I^i are the sets of inequality constraints that are not active in step 2.

(b) Eliminate n_A^i control variables from the vector $z^i = \begin{bmatrix} z_A^i \\ z_I^i \end{bmatrix}$, so as to define

$$z_R^i = z_I^i, \quad x_R^i = \begin{bmatrix} x^i \\ z_A^i \end{bmatrix}, \quad i = 1,2,\ldots N \tag{2.9}$$

where z_R^i is the redefined vector of control variables which result from eliminating the vector z_A^i of n_A^i elements, and x_R^i is the expanded vector of state variables.

Step 4 - Solve the restricted problem:

$$\begin{aligned} \underset{d,z_R^i,z_R^i,\ldots z_R^i}{\text{minimize}} \quad & C = C^o(d) + \sum_{i=1}^{N} C^i(d,z_R^i,x_R^i) \\ \text{s.t.} \quad & \left.\begin{aligned} h_R^i(d,z_R^i,x_R^i) &= 0 \\ g_R^i(d,z_R^i,x_R^i) &\leq 0 \end{aligned}\right\} \; i=1,2,\ldots N \\ & f(d) \leq 0 \end{aligned} \tag{2.10}$$

Step 5 - Return to step 2 and iterate until

(a) no further changes occur in the values of the variables d, or

(b) the same set of inequality constraints become active again, in step 2.

Note that in step 4 the projection-restriction strategy really consists in solving problem (2.1) simultaneously for all variables, but in general with a smaller number of decision variables, since many of these get eliminated by the active constraints determined in step 2. Clearly, the effectiveness of this strategy relies heavily on the number of inequality constraints that actually become active at the solution.

Grigoriadis (1971) and Ritter (1973) found that in their problems relatively few inequality constraints in the subproblems would become active. Therefore, they proposed to eliminate all the variables z^i, $i=1,2,\ldots,N$, in step 3, Grigoriadis (1971) with the use of the pseudo-inverse of the corresponding matrix of z^i, and Ritter (1973) with a square matrix which was generated when solving the subproblems. Unfortunately these techniques cannot be extended readily to the case when constraints are nonlinear, since they rely heavily on the assumption of linearity of the constraints. However, the basic idea of the projection-restriction strategy can indeed be extended to the problem of designing flexible chemical plants.

An examination of the results for flexible plants obtained by Grossmann and Sargent (1978, 1979) shows that a surprisingly large number of inequality constraints are actually active at the solution, as can be seen in Table 2.1. The main reason for this appears to be the monotonicity of the cost function (in the constraint space) which seems to be characteristic of design problems. Since in general one can expect to have a large number of active constraints at the solution, it clearly suggests that the projection-restriction strategy can greatly simplify solving problems of the type as formulated in (2.1). However, for successful application of the projection-restriction strategy there are two

Table 2.1: Active constraints at the solution, in the design problems solved by Grossmann and Sargent (1978, 1979)

Problem	Number of decision variables	Number of inequality constraints	number of *active constraints* at the solution
Pipeline	8	20	5
Multiproduct Batch Plant			
(a) problem 1a.	10	23	11 *
(b) problem 2.	14	39	14
Reactor-Separator System	7	24	4
Heat Exchange Network	15	65	15

* One of the constraints was redundant at the solution

problems that have to be considered. The first one is *finding an initial feasible point* in step 1. The second is a *procedure for the elimination of variables* in step 3, which avoids singularities in the system of equations. These are discussed in the following sections.

2.5 FINDING A FEASIBLE POINT

The problem of finding a feasible point for a design problem is in general a nontrivial task, because of the nonlinearities involved. In problem (2.1) the main difficulty when using the projection-restriction strategy consists in finding a value of d such that feasible solutions exist for the subproblems in (2.2). One approach to find a feasible point is to replace the cost function in (2.1) by the sum of squares of deviations of the violated constraints, thus leading to the nonlinear program:

$$\begin{aligned}
&\underset{d,z^1,z^2,\dots z^N}{\text{minimize}} \quad \Phi = \sum_{i=1}^{N}\sum_{j=1}^{M}\left[\max\left\{0,\ g_j^i(d,z^i,x^i)\right\}\right]^2 \\
&\text{s.t.} \quad \left.\begin{aligned} h^i(d,z^i,x^i) &= 0 \\ g^i(d,z^i,x^i) &\le 0 \end{aligned}\right\} \ i=1,2,\dots N \\
&\qquad\ f(d) \le 0
\end{aligned} \tag{2.11}$$

This problem can be handled by an NLP algorithm based on an active set strategy for the constraints as the one developed by Sargent and Murtagh (1973). Since the objective function in (2.11) has discontinuous second order derivatives the optimization should be performed with the steepest descent direction in the constraint space. As this procedure does not require an estimation of the inverse of the Hessian matrix, storage requirements can be reduced.

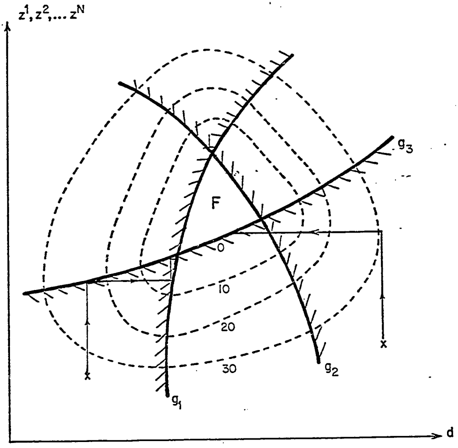

Figure 2.2: Contours of the objective function in problem (2.11)

Although solving (2.11) simultaneously for all variables works very well for relatively small problems, it may be desirable to use a decomposition scheme for large problems. One alternative is to use the steps similar to those in the feasible decomposition strategy, with the objective function as given in (2.11) above. Hence it consists of the following steps:

Step 1 - Guess a starting point d, z^i, x^i, $i = 1,2,....N$.

Step 2 - By keeping the vector d fixed, solve the N subproblems (that is, for $i = 1,2,....N$):

$$\begin{aligned} \underset{z^i}{\text{minimize}} \quad & \Phi^i = \sum_{j=1}^{M} \left[\max \left\{ 0,\ g_j^i(d,z^i,x^i) \right\} \right]^2 \\ \text{s.t.} \quad & h^i(d,z^i,x^i) = 0 \\ & g^i(d,z^i,x^i) \leq 0 \end{aligned} \tag{2.12}$$

Step 3 - By keeping the vectors z^i, $i=1,2,....N$ fixed, solve the problem:

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & \Phi = \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\max \left\{ 0,\ g_j^i(d,z^i,x^i) \right\} \right]^2 \\ \text{s.t.} \quad & \left. \begin{aligned} h^i(d,z^i,x^i) &= 0 \\ g^i(d,z^i,x^i) &\leq 0 \end{aligned} \right\} i=1,2,....N \\ & f(d) \leq 0 \end{aligned} \tag{2.13}$$

Step 4 - If convergence is not achieved ($\Phi > 0$) return to step 2.

It is observed that unlike in the case of finding the optimal solution

of (2.1), the convergence of this method to find a feasible point is quite good. The reason for this is that the NLP defined by (2.11) could have an infinite number of minima when the feasible region is non-empty. In this case the objective function defines a plateau of zero-value for the feasible region as shown in Figure 2.2, and hence there is usually no problem of slow convergence in the neighborhood of a feasible solution. Also, note in Figure 2.2 that outside the feasible region the contours of Φ in (2.11) are quadratic in the constraint functions so that the objective function will tend to be well behaved.

2.6 VARIABLE ELIMINATION IN THE RESTRICTION STEP

The elimination of variables in step 3 of the projection-restriction strategy is performed for each period i by including the active inequality constraints in the set of equations. This implies that from the state and control variables x^i, z^i a new set of control variables z_R^i must be determined, and that the sequence of calculation for the new set of equations h_R^i has to be derived, for each period $i = 1,2,....N$.

Since a number of algorithms are available for selecting decision variables and determining sequences of calculation for rectangular systems (see e.g. Lee et al., 1966; Christensen and Rudd, 1969; Edie and Westerberg, 1971; Leigh, 1973; Stadtherr et al., 1974; Book and Ramirez, 1976; Hernandez and Sargent, 1979), it would seem that they could be applied without difficulty in the problem of designing flexible chemical plants. It must be pointed out, however, that difficulties may arise when deriving the solution procedure for the restricted problem (2.10), since the added inequality constraints can lead to *redundant or inconsistent* equations, and hence, produce a *singular* system of equations. Therefore, these algorithms must be extended according to the following procedure for eliminating the variables in each i in the restricted problem:

Step 1 - Add all the active constraints $g_A^i(d,z^i,x^i) = 0$, to the system of equations $h^i(d,z^i,x^i) = 0$, thus giving rise to a new system of equations $h_R^i(d,z^i,x^i) = 0$.

Step 2 - Perform the optimal reordering of the new system of equations $h_R^i = 0$, by minimizing the number of torn variables in z^i, x^i.

Step 3 - Select control variables z_R^i as decision variables, and delete equations if necessary so as to obtain a non-singular square system of equations.

It should be noted that due to the reordering of variables in step 2, the vector z_R^i can in fact contain some of the state variables from x^i. Also, it is essential to keep the design variables d as decision variables throughout, and not to force them to become either state or torn variables during the reordering. Again, in step 2 above, since it is possible that the resulting system has more equations than variables, a suitable equation ordering algorithm must be used, as for instance the one by Leigh (1973). The optimal sequence determined with such an algorithm is one where the system $h_R^i(d,z^i,x^i) = 0$, is reordered as shown in Figure 2.3 in two sets of equations:

$$\begin{aligned} s(u,v) &= 0 \\ r(u,v) &= 0. \end{aligned} \tag{2.14}$$

Here the subsystems s and r form a partition of the vector of equations h_R^i, whereas the vectors of variables u and v are a partition of the vector $[(z^i)^T, (x^i)^T]^T$. As shown in Figure 2.3, s is the set of non-recycle equations with lower-triangular structure, which can be solved

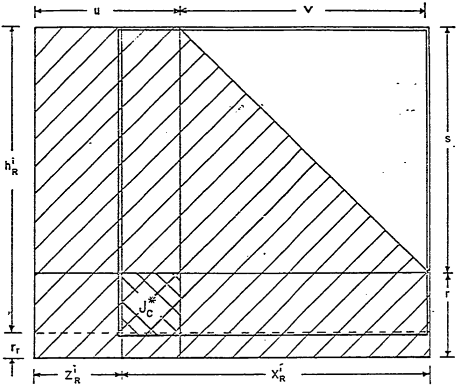

Figure 2.3: Structure resulting from equat... ...dering

sequentially for the vector v given a value of u, and r corresponds to the set of recycle equations. Since v can be treated as an implicit function of u, the system of equations in (2.14) can be reduced to the form:

$$r(u, v(u)) = 0 \tag{2.15}$$

The vector u represents the set of decision variables z_R^i and torn variables t^T, that is, $u^T = [(z_R^i)^T, t^T]$ and for the optimal sequence its dimensionality is at a minimum.

In order to delete the appropriate equations in step 3, it is sufficient to choose the largest nonsingular subset of equations at the current point. Note that for a fixed u, the subsystem s in (2.14) can be assumed to be nonsingular due to its lower-triangular structure, and the singularity of the system (2.14) can be analyzed through the jacobian $J_c(r,u)$ of (2.15) which is given by

$$J_c(r,u) = \frac{\partial r}{\partial u} - \frac{\partial r}{\partial v}\left(\frac{\partial s}{\partial v}\right)^{-1}\frac{\partial s}{\partial u} \tag{2.16}$$

This jacobian matrix can be evaluated numerically at the current point by performing perturbations in the vector u. To determine the equations to be deleted the following procedure can be followed. The square submatrix J_c^* of highest rank is obtained by performing a Gaussian elimination on the jacobian matrix J_c in (2.16). The variables in u that correspond to the columns of the submatrix J_c^* will be chosen as torn variables t. The remaining variables in u will correspond to the decision variables z_R^i. Those equations r_R in r that are not included in the rows of the submatrix J_c^*, will be deleted and treated as inequality constraints. In this way, the jacobian matrix of the resulting system of equations can be ensured to be of full rank and hence non-singular. Also, note that the jacobian matrix J_c to be analyzed is usually of much smaller size than the jacobian of the system in (2.14).

A detailed discussion on the procedure for the selection of control variables in the restriction step is given in the Appendix, where it is presented in a more general context of selection of decision and torn variables in process design calculations. It is useful to note that this procedure can also be applied in the initial stage of formulating the design problem (2.1), in order to identify the proper choice for the decision variables in the design problem and also to detect any redundancy or inconsistency in the system. This is an important aspect in formulation of problems for the design of flexible chemical plants since one has to be careful to maintain a positive number of degrees of freedom, and also have a consistent system.

The procedure indicated above for finding an initial feasible point and for the variable elimination in the restriction step, complete the algorithm required for the projection-restriction strategy. The next chapter presents a discussion on the expected performance of the proposed decomposition strategy for solving deterministic multiperiod problems, and an example problem which illustrates the application and confirms the expected performance of the projection-restriction strategy.

CHAPTER 3

NUMERICAL PERFORMANCE OF THE PROJECTION-RESTRICTION STRATEGY

The projection-restriction strategy presented in the previous chapter is expected to be very efficient, provided that a large fraction of the original inequality constraints actually become active at the solution of the design problem. This is in fact true in most of the design problems, as shown in Table 2.1, thus making the above strategy an effective decomposition scheme in solving large multiperiod problems that arise in the design of flexible chemical plants. This chapter presents an analysis of the relative gain in computational effort obtained by using the proposed decomposition strategy, and the effect of the size of the design problem as well as the number of active constraints on the expected numerical performance. An example problem has also been solved to illustrate the application of the projection-restriction strategy, and to observe the actual performance.

3.1 ANALYSIS OF THE EXPECTED NUMERICAL PERFORMANCE

The relative advantage in using a decomposition technique may in general depend on the size of the problem being solved, and hence it is appropriate to analyze the characteristics of the proposed decomposition strategy from this viewpoint.

Based on some simple assumptions, a relationship can be derived between the CPU-time and the size of the design problem in terms of number of periods.

Let n_d be the number of design variables, n_z the number of control variables in each period and N the number of periods considered in the design problem. Assume that the CPU-time t_P for the projection step is given by

$$t_P = a_P \left(n_z\right)^p \tag{3.1}$$

If α, $0 \leq \alpha \leq 1$, is the average fraction of the control variables remaining in the restriction problem, then the CPU-time t_R for the restriction step can be expressed as

$$t_R = a_R \left(n_d + N \alpha n_z\right)^r \tag{3.2}$$

Also, let the CPU-time t_Q for solving the problem without decomposition be

$$t_Q = a_Q \left(n_d + N n_z\right)^q \tag{3.3}$$

If K is the number of iterations (passes) through the projection (and restriction) steps that is required for convergence, then the total CPU-time needed to solve the design problem using the decomposition strategy is

$$t_{PR} = K N a_P \left(n_z\right)^p + K a_R \left(n_d + N \alpha n_z\right)^r \tag{3.4}$$

In general the exact values of p, q, r and a_P, a_Q, a_R, K depend on the particular problem at hand. However, for a given problem the values of a_P, a_Q, a_R can be expected to be of the same order of magnitude, and for a gradient based non-linear programming algorithm one can expect to have the values of p, q and r to lie between 2 and 3. Also, the value of K is likely to be small.

If all the control variables are eliminated in the restriction step, $\alpha = 0$ and from (3.4) it is clear that the CPU-time t_{PR} is linear in N the number of periods, as given in (3.5) below:

$$\alpha = 0: \qquad t_{PR} = K\, N\, a_P \left(n_z\right)^p + K\, a_R \left(n_d\right)^r \tag{3.5}$$

For $\alpha > 0$ the relative savings in computational time in using the decomposition strategy can be determined from (3.3) and (3.4),

$$\frac{t_{PR}}{t_Q} = \frac{K\, N\, a_P \left(n_z\right)^p + K\, a_R \left(n_d + N\, \alpha\, n_z\right)^r}{a_Q \left(n_d + Nn_z\right)^q} \tag{3.6}$$

Since a_P, a_R, a_Q are of the same order of magnitude; $0 \leq \alpha < 1$, and $p, q, r > 1$, it is clear from (3.6) that the relative advantage in using the decomposition strategy is enhanced by larger values of N and smaller values of α. In fact, for a given value of N there is a threshold value α^t, below which savings in CPU-time can be ensured by using the decomposition strategy. This threshold value determines a useful range $0 \leq \alpha \leq \alpha^t$, which can be determined from (3.6) with $t_{PR} \leq t_Q$, thus obtaining α^t as

$$\alpha^t = \frac{1}{Nn_z}\left[\left\{ \left(\frac{a_Q}{a_R K}\right)\left(n_d + Nn_z\right)^q - \left(\frac{Na_P}{a_R}\right)\left(n_z\right)^p \right\}^{1/r} - n_d \right] \tag{3.7}$$

Assuming $a_P = a_R = a_Q$ and $p = r = q$, (3.7) can be simplified as

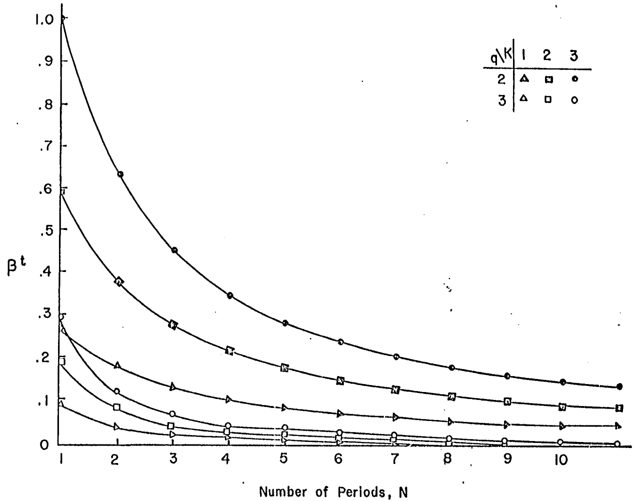

Figure 3.1: Minimum fraction β^t of control variables that must be eliminated as given in (3.8)

$$\alpha^t = \left[\frac{1}{K} \left\{ \frac{n_d}{Nn_z} + 1 \right\}^q - \frac{1}{N^{q-1}} \right]^{1/q} - \frac{n_d}{Nn_z} = 1 - \beta^t \qquad (3.8)$$

where β^t indicates the minimum fraction of control variables to be eliminated in the restriction problem. Figure 3.1 shows some plots of β^t versus N as given by the expression in (3.8) above for the case when $n_d = n_z$. There are two sets of three plots, for q = 2, 3 and K = 1, 2, 3. As can be seen from this Figure 3.1, β^t is smaller for larger values of q and for smaller values of K. Also, for a given q and K, β^t decreases rapidly with N even for relatively small values of N, and approaches zero asymptotically for large N.

As an example, take the case when q = 2 and when a single pass (K = 1) in the decomposition strategy can lead to the optimal solution. If the design problem involves five periods, only 8% of the control variables must be eliminated to achieve a relative gain in CPU-time with the decomposition strategy. These percentages increase to 18% for K = 2 and 28% for K = 3. For a ten period problem the percentages reduce respectively to 4%, 10% and 15% for K = 1,2 and 3. Thus, in general one can expect to obtain significant gains in computational time when solving multiperiod design problems with the projection-restriction strategy, even when the number of active constraints is not very large.

3.2 NUMERICAL EXAMPLE

To confirm the expected performance of the proposed projection-restriction strategy, an example problem is solved. The flowsheet consists of a reactor and a heat exchanger as shown in Figure 3.2. The reaction is assumed to be first order exothermic, of the type A → B. The flowrate through the heat exchanger loop is adjusted to maintain the reactor temperature below T^i_{1max} as given in Table 3.1 and to get a minimum of 90% conversion.

This plant is to be designed so as to produce different products in N successive periods within each year. The performance equations of such a system, for any period i, i = 1, 2, ... N, are as follows:

Reactor, material balance:

$$F^i_o \left(C^i_{Ao} - C^i_{A1} \right) / C^i_{Ao} = V^i k^i_o \exp(-E^i/RT^i_1) C^i_{A1} \tag{3.9}$$

Reactor, heat balance:

$$(-\Delta H)^i_{rxn} F^i_o \left(C^i_{Ao} - C^i_{A1} \right) / C^i_{Ao} = F^i_o C^i_p (T^i_1 - T_o) + Q^i_{HE} \tag{3.10}$$

Heat exchanger, heat balances:

$$Q^i_{HE} = F^i_1 C^i_p (T^i_1 - T^i_2) \tag{3.11}$$

$$Q^i_{HE} = F^i_w C^i_{pw} (T^i_{w2} - T^i_{w1}) \tag{3.12}$$

Heat exchanger, design equations:

$$Q^i_{HE} = A\ U\ (\Delta T)^i_m \tag{3.13}$$

$$(\Delta T)^i_m = \frac{\left(T^i_1 - T^i_{w2}\right) - \left(T^i_2 - T^i_{w1}\right)}{\log\left\{ \left(T^i_1 - T^i_{w2}\right) / \left(T^i_2 - T^i_{w1}\right) \right\}} \tag{3.14}$$

The data for the problem are given in the Table 3.1. The design problem corresponds to an optimization problem with the equality constraints given by the performance equations above, and the following inequality constraints for each period i = 1, 2, ... N.

$$\hat{V} \geq 0 \tag{3.15}$$

$$A \geq 0 \tag{3.16}$$

$$V^i \geq 0 \tag{3.17}$$

$$\hat{V} - V^i \geq 0 \tag{3.18}$$

$$F^i_w \geq 0 \tag{3.19}$$

$$F^i_1 \geq 0 \tag{3.20}$$

$$0.9 \leq (C^i_{Ao} - C^i_{A1})/C^i_{Ao} \leq 1.0 \tag{3.21}$$

$$T^i_1 \leq T^i_{1max} \tag{3.22}$$

$$T^i_1 - T^i_2 \geq 0 \tag{3.23}$$

$$T_{w1} \leq T^i_{w2} \leq 356 \tag{3.24}$$

$$T^i_1 - T^i_{w2} \geq 11.1 \tag{3.25}$$

$$T^i_2 - T_{w1} \geq 11.1 \tag{3.26}$$

The objective function being minimized is the total annual cost (\$/yr),

$$C = 691.2\ \hat{V}^{0.7} + 873.6\ A^{0.6} + \sum_{i=1}^{N} \omega^i \left(1.76\ F^i_w + 7.056\ F^i_1\right) \tag{3.27}$$

where t^i corresponds to the number of hours of operation for each period i=1,2,....N, in one year. The objective function includes the investment cost of the reactor and heat exchanger, and the operating cost of the cooling water and recycle.

There are $2 + 9N$ variables $\hat{V}$, A, C^i_{A1}, T^i_1, T^i_2, T^i_{w2}, F^i_1, F^i_w, V^i, $(\Delta T)^i_m$, Q^i_{HE}, i=1,2,....N; 6N equations and $2 + 10N$ inequality constraints and bounds, for a problem with N different periods. This gives rise to $2 + 3N$ degrees of freedom for the design problem. Selecting as decision variables the design variables $\hat{V}$, A, and the control variables T^i_1, T^i_2, T^i_{w2},

$i=1,2,....N$, the sequence of calculation for the equations in each period is given in Table 3.2. The corresponding nonlinear program consists of 2 + 3N decision variables, 6N nonlinear inequality constraints and N linear inequality constraints. Note that (3.15), (3.16), (3.22), (3.24) and (3.26) are simple bounds on the decision variables, that (3.23) is a linear inequality and the remaining constraints are nonlinear.

The problem has been solved for five cases corresponding to N=1,2,3,4,5. In each case, the plant is designed to produce N different products that have different feed rates, concentrations, reaction rate constants, etc. as indicated in Table 3.1. In all the five cases, when solving the subproblems in the projection step it is found that constraint (3.21) is active at its lower bound, and constraints (3.22) and (3.24) are active at their upper bounds, for all periods. Adding these active constraints to the equations in Table 3.2, the variables T_1^i, T_2^i, T_{w2}^i, $i=1,2,...N$ were eliminated by ordering the new system of equations. This gives rise to only two decision variables $\hat{V}$ and A for the restricted problem, as shown in Table 3.3.

The starting point given in Table 3.1, which is infeasible, is used for all five cases. The initial feasible points used in the projection-restriction strategy were obtained by minimizing alternately with respect to d and z^i, $i=1,2,....N$, the sum of squares of deviations of violated constraints. The optimizations were performed using the variable metric projection method (Sargent and Murtagh, 1973), and the solutions were obtained with a tolerance of 10^{-2} for the norm of the gradient of the objective function projected in the constraint space.

The optimal sizes of the reactor and the heat exchanger are presented in Table 3.4 for the five cases. The formulation of the problem itself ensures that these optimal designs are flexible, as they meet the specifications for the various products involved at a minimum annual cost.

The computing requirements for finding the initial feasible points are shown in Table 3.5. Here, it was found that optimizing alternately for d and z^i is more efficient than considering all these variables simultaneously, particularly when the number of periods is large. However, the more significant gains in computational requirements are achieved when the projection-restriction strategy is applied, once the problem becomes feasible. Table 3.6 gives the CPU-time requirements for the projection and restriction steps. Table 3.7 and Figure 3.3 give a comparison of the computational requirements in solving the design problem with and without the use of the decomposition scheme. A striking feature in the performance of the proposed decomposition strategy is that the CPU-time increases only linearly with the size of the design problem. It is interesting to note that the design problem for the five-period case was solved by using the proposed decomposition strategy in only 31.4 sec which is about the same time required for the one-period problem without using any decomposition.

3.3 DISCUSSION

The results of the above example show that the performance of the proposed decomposition strategy for the design of flexible chemical plants is very encouraging. In fact, the decomposition strategy has been applied to solve the design problems in Chapter 5 as well, requiring only moderate amount of computational time. It is to be noted in Table 3.7, that the computational effort required to solve the design problem of this chapter has been reduced by an *order of magnitude* with the application of the *projection-restriction strategy*. An important trend in the results is that the computational time required is approximately linear with the size of the design problem (number of periods). This suggests that the reduction in computational effort with the proposed method is even more dramatic in larger problems. This is to be expected, since experience with different nonlinear programming

algorithms indicate that they are much more likely to be successful in converging to the optimal solution when the number of decision variables is relatively small. Also, it will lead to a more reliable method in obtaining the desired optimum design solutions, since the large nonlinear programming problem is reduced to a sequence of smaller subproblems.

Note that the proposed decomposition strategy is conceptually quite general, and does not presuppose the use of any particular optimization algorithm for solving the resulting nonlinear programming (NLP) problems. Although no numerical experimentation has been performed with different NLP algorithms, it is clear from the derivations given at the beginning of this chapter, that one can expect significant savings in computational effort with almost any algorithm for solving the nonlinear program. The exponential increase in CPU-time with the increase in number of periods that is observed for solving the example problem *without decomposition* is typical of any optimization algorithm. In fact, the above example problem has been solved for the case of 1, 2 and 3 periods (without the application of any decomposition strategy) using the Han-Powell Algorithm (Powell, 1977) for optimization; requiring 14.1sec, 126.5sec and 543.3sec respectively to attain the solution (as against 32.2sec, 176.2sec and 479.6sec respectively when using the variable metric projection algorithm). These computational results seem to compare with those obtained by using the variable metric projection algorithm of Sargent and Murtagh (1973), although in general the latter was more accurate because of its stringent termination criteria - for any given error tolerance. From these results it can be seen that whatever be the optimization algorithm used, the application of the decomposition strategy will certainly enhance the efficiency in solving the multiperiod design problems.

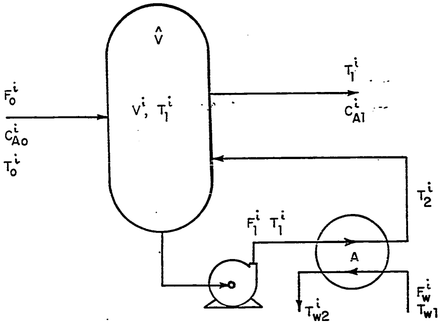

Figure 3.2: Flowsheet for the Example Problem

Table 3.1: Data for the Example Problem

Period i:	1	2	3	4	5	
(E^i/R)	555.6	583.3	611.1	527.8	500.0	K
$(-\Delta H)^i_{rxn}$	23260	25581	27907	20930	18604	kJ/kgmole
k^i_o	0.6242	0.6867	0.7491	0.5619	0.4994	hr^{-1}
C^i_p	167.4	188.4	209.3	146.5	125.6	kJ/kgmole K
C^i_{Ao}	32.04	40.05	48.06	24.03	32.04	$kgmole/m^3$
F^i_o	45.36	40.82	36.29	49.90	54.43	kgmole/hr
T^i_{1max}	389.0	383.0	378.0	394.0	400.0	K
U	1635.34	1635.34	1635.34	1635.34	1635.34	kJ/m^2 hr K
T_o	333.3	333.3	333.3	333.3	333.3	K
T_{w1}	300.0	300.0	300.0	300.0	300.0	K

Starting Point:	$\hat{V}$	=	14.16	m^3	
	A	=	11.15	m^2	
	T^i_1	=	367	K	
	T^i_2	=	328	K	i=1,2,.....N
	T^i_{w2}	=	333	K	

Table 3.2: Equation Ordering in the Projection Step

VARIABLES

	$\hat{V}$	A	T_1^i	T_2^i	T_{w2}^i	$(\Delta T)_m^i$	Q_{HE}^i	C_{A1}^i	V^i	F_1^i	F_w^i
14			X	X	X	X					
13		X				X	X				
10			X				X	X			
9			X					X	X		
11			X	X			X			X	
12					X		X				X

Table 3.3: Equation Ordering in the Restriction Step

V A R I A B L E S

	$\hat{V}$	A	C^i_{A1}	T^i_1	T^i_{w2}	V^i	Q^i_{HE}	$(\Delta T)^i_m$	T^i_2	F^i_1	F^i_w
21			X								
22				X							
24					X						
9			X	X		X					
10			X	X			X				
13		X					X	X			
14				X	X			X	X		
11				X			X		X	X	
12					X		X				X

Table 3.4: Solution of the Exampie Problem

Number of Periods N *	V m^3	A m^2	Optimum Annual Cost $/yr
1	5.318	7.562	0.980×10^4
2	5.318	8.417	1.010×10^4
3	5.318	9.513	1.042×10^4
4	7.915	9.262	1.096×10^4
5	7.915	9.095	1.080×10^4

* Number N indicates periods 1,2,....N taken together.

Table 3.5: Computational results for finding an initial feasible point, CPU-Time (DEC-20):

Number of Periods N *	Optimize all Variables Simultaneously	Optimize Alternatively d and z
1	0.228	0.204
2	0.371	0.432
3	3.745	0.670
4	1.763	0.878
5	2.954	1.083

* Number N indicates periods 1,2,....N taken together.

Table 3.6: Computational results for the Projection-Restriction Algorithm, CPU-Time (DEC-20):

Number of Periods N *	Projection	Restriction	Total CPU-Time DEC-20 sec
1	4.531	1.304	5.835
2	12.129	2.221	14.350
3	15.918	1.416	17.334
4	21.998	2.181	24.179
5	27.311	3.005	30.316

* Number N indicates periods 1,2,....N taken together.

Table 3.7: Computational results for solving the design problem, CPU-Time (DEC-20):

Number of* Periods N	Number of Decision Variables	Number of Inequality Constraints	Computational Time without	with Decomposition
1	5	7	32.2	6.0
2	8	14	176.2	14.8
3	11	21	479.6	18.0
4	14	28	---	25.0
5	17	35	---	31.4

* Number N indicates periods 1,2,....N taken together.

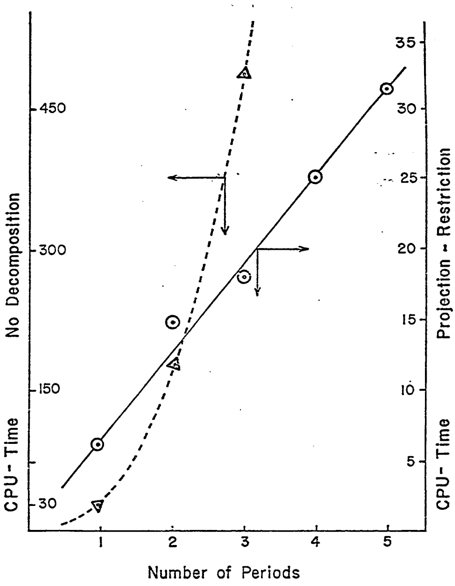

Figure 3.3: Computational time for solving the design problem versus the size of the problem

3.4 NOMENCLATURE FOR THE NUMERICAL EXAMPLE PROBLEM

A	:	Heat transfer area of the heat exchanger, m^2
C^i_{Ao}	:	Concentration of reactant in the feed stream,kgmole/m^3
C^i_{A1}	:	Concentration of reactant in the product stream,kgmole/m^3
C_p	:	Heat capacity of reaction mixture, kJ/kgmole K
(E^i/R)	:	Ratio of activation energy to gas constant, K
F^i_o	:	Feed flowrate, kgmole/hr
F^i_1	:	Flowrate of the recycle, kgmole/hr
F_w	:	Flowrate of the cooling water, kg/hr
$(\Delta H)^i_{rxn}$	:	Heat of reaction, kJ/kgmole
k_o	:	Arrhenius rate constant for reaction, hr^{-1}
N	:	Total number of periods considered for design, (integer)
Q^i_{HE}	:	Heat exchanger load, kJ/hr
T_o	:	Temperature of feed stream, K
T^i_1	:	Reaction temperature, K
T^i_2	:	Recycle temperature, K
T_{w1}	:	Inlet temperature of cooling water, K
T_{w2}	:	Outlet temperature of cooling water, K
$(\Delta T)^i_m$	:	Mean temperature difference, K
U	:	Overall heat transfer coefficient, kJ/m^2 hr K
$\hat{V}$	:	Reactor volume (design capacity), m^3
V^i	:	Reaction volume in the operating period i, m^3
ω^i	:	weighting factor for the period i.

Superscript i refers to the period of operation, i=1,2,....N

CHAPTER 4

OPTIMAL PROCESS DESIGN UNDER UNCERTAINTY

In the optimal design of chemical processes it is very often the case that considerable uncertainty exists in the value of some of the parameters. For instance, these parameters may correspond to internal process parameters such as transfer coefficients, reaction constants, efficiencies or physical properties. In addition, the uncertain parameters may also include those external to the process itself, such as specifications in the feed streams, utility streams, environmental conditions or cost data. The values of these parameters are either known only approximately during the design stage, or they may actually vary or fluctuate during the operation of the plant. Hence, in designing an optimal chemical process system, these parameters have to be treated as having uncertain values, and the effect of these uncertainties on both the *optimality* and the *feasibility of operation* must be considered.

To account for the uncertainties in the values of these parameters, normally the procedure that is used in practice is to assume some nominal values for the parameters, and then apply empirical overdesign factors to the resulting design. Since this procedure lacks a firm rational basis, a number of different methods have been suggested to account for the uncertainties in a

more systematic manner. These methods differ mainly in the basic design strategies that are postulated, since in principle the problem of design under uncertainty is not well-defined. A brief review of the different approaches and their shortcomings is given in Chapter 1.

As mentioned in Chapter 1, *the main objective in the problem of optimal process design under uncertainty is to guarantee feasible steady state operation of the plant under the various different conditions that may be encountered, so as to meet the specifications.* In other words, while minimizing the cost to select the optimal design, the main concern of a design engineer must be to ensure that even when the parameters take on different values within a given bounded region, suitable manipulation in the operation of the plant can be achieved in order to satisfy the specifications. Grossmann and Sargent (1978) have proposed a design strategy which tries to incorporate this objective by designing a plant flexible enough to accommodate the uncertainty in the values of the parameters. The basic idea in this strategy is to take advantage of the fact that control variables can be adjusted to satisfy the design specifications during the operation of the plant, as it is only the design of the plant itself that will remain fixed. Based on this strategy, a rigorous mathematical formulation of the problem of optimal design of flexible chemical plants with uncertainty in parameter values is given in this chapter. As will be shown in Chapter 5, this formulation is more general than the one presented by Grossmann and Sargent (1978) which can fail in some cases to guarantee feasibility. The interpretations and properties of the proposed formulation for the case of nonlinear convex functions are also presented here, whereas the proposed solution algorithms with numerical examples are given in Chapter 5.

4.1 MATHEMATICAL FORMULATION

The variables in the design problem of a chemical plant with uncertainty in parameter values can be partitioned into four categories. The vector d of *design variables* is associated with the sizing of the units. These remain fixed once the design is implemented, and do not vary with the changes in the operation of the plant. The vector z denotes the *control variables*, the values of which can be chosen during the operation so as to meet the specifications and also to minimize the operating cost. The vector x corresponds to the *state variables* which are determined by solving the set of equations representing the process system. Finally, θ is the vector of *independent parameters* in the design whose values are subject to uncertainty. Assuming that *bounded* range of values are specified for these parameters, the region T that is defined to contain all possible values of these parameters is given by

$$T = \left\{ \theta \mid \theta^L \le \theta \le \theta^U \right\} \tag{4.1}$$

where θ^L and θ^U represent given lower and upper bounds on θ. We assume that the control variables z lie in a compact set.

In order to derive the mathematical formulation, it is convenient to consider the design strategy as being composed of two stages: an *operating stage* and a *design stage*.

(1) **Operating stage:** Assuming that a given design d has been selected, it is anticipated that the plant will be operated optimally while satisfying the constraints of the process for all possible realizations of the parameters θ within the region T. Hence, the objective in this stage is to select for every realization $\theta \in T$, a control z which is optimal and feasible.

Clearly, for the given d and for any value of θ, the state variables can be expressed as an implicit function of the control variables z, from the system of equations of the process,

$$h(d,z,x,\theta) = 0 \quad \Rightarrow \quad x = x(d,z,\theta) \tag{4.2}$$

Since the control variable z should be selected so as to satisfy the specifications given by the vector of inequality constraints,

$$g(d,z,x,\theta) \;=\; g(d,z,x(d,z,\theta),\theta) \;=\; f(d,z,\theta) \;\leq\; 0 \tag{4.3}$$

the optimal feasible operation of the plant that minimizes the cost will be given by the nonlinear program

$$\begin{aligned} &\underset{z}{\text{minimize}} && C(d,z,\theta) \\ &\text{s.t.} && f(d,z,\theta) \;\leq\; 0 \end{aligned} \tag{4.4}$$

Since the control variables z lie in a compact set, the minimum is attained. The solution of this problem defines a cost function $C^*(d,\theta)$ which corresponds to the optimal operation of the plant for fixed values of d, θ. Moreover, if the optimization is performed for every realization $\theta \in T$, the average cost of operation will be given by the expected value $\underset{\theta \in T}{E}\left\{C^*(d,\theta)\right\}$.

(2) **Design stage:** In order to achieve the basic objective of feasibility of operation in the region T of parameters, the design variable d must be chosen so as to ensure that for every value $\theta \in T$ the control variable z can be selected to satisfy the constraints in (4.4). Note that an improper selection of d can lead to infeasible operation for some realizations of θ in which case no selection of the control z will exist so as to satisfy the inequality constraints in (4.4). Furthermore, in order to achieve the optimal design, the design d must be selected so as to minimize the expected value of the optimal cost function $C^*(d,\theta)$ over the entire region T.

This strategy can then be expressed mathematically as

$$\begin{aligned} &\underset{d}{\text{minimize}} \quad \underset{\theta \in T}{E} \left\{ C^{*}(d,\theta) \right\} \\ &\text{s.t.} \qquad \forall\ \theta \in T \left\{ \exists\ z \left(\forall\ j \in J \left[f_j(d,z,\theta) \leq 0 \right] \right) \right\} \end{aligned} \tag{4.5}$$

where $J = \{1,2,....m\}$ is the index set for the components of vector f. The constraint in (4.5) is denoted as the *feasibility constraint*, because the existence of a feasibility region of operation for every $\theta \in T$ can be ensured if and only if this constraint is satisfied. In fact, this logical constraint states that for every point $\theta \in T$ in the space of parameters, there must exist at least one value for the vector z of control variables that gives rise to non-positive values for all the individual constraint functions. Qualitatively, this means that irrespective of the actual values taken by the parameters, with the selected design d, the plant can be operated to satisfy the specifications.

Since the objective function in (4.5) is itself determined through the NLP in (4.4), the problem of optimal design under uncertainty can be formulated in its final form as a *two-stage programming problem*:

$$\begin{aligned} &\underset{d}{\text{minimize}} \quad \underset{\theta \in T}{E} \left\{ \min_{z}\ C(d,z,\theta) \mid f(d,z,\theta) \leq 0 \right\} \\ &\text{s.t.} \qquad \forall\ \theta \in T \left\{ \exists\ z \left(\forall\ j \in J \left[f_j(d,z,\theta) \leq 0 \right] \right) \right\} \end{aligned} \tag{4.6}$$

Note that since there are an infinite number of possible realizations for the values of the parameters θ, and since the optimal operation of the plant is implicitly dependent on θ, the overall number of decision variables involved in problem (4.6) is infinite. This is because for every value of θ an optimal value of the control variables z is being chosen. Also, note that the feasibility constraint represents an infinite set of constraints since the

inequalities in (4.4) are defined for the infinite set of values $\theta \in T$. Therefore, problem (4.6) corresponds to a *two-stage nonlinear infinite program* (NLIP).

4.2 SIMPLIFICATION OF THE TWO-STAGE NLIP

The two-stage nonlinear infinite program in (4.6) that represents the mathematical formulation of the design problem under uncertainty poses great computational difficulties for numerical solution, and in fact has a more complex structure than the semi-infinite programs treated in the literature (see Hettich, 1979). A first step in simplification so as to make the problem more tractable, is to perform a discretization over the parameter space in order to approximate the expected cost by a weighted average (Grossmann and Sargent, 1978), which reduces (4.6) to:

$$\begin{aligned} &\underset{d,z^1,z^2,\ldots,z^N}{\text{minimize}} \quad \sum_{i=1}^{N} \omega^i \, C(d,z^i,\theta^i) \\ &\text{s.t.} \quad f(d,z^i,\theta^i) \le 0, \quad i = 1,2,\ldots,N \\ &\qquad \forall\, \theta \in T \left\{ \exists\, z \left(\forall\, j \in J \left[f_j(d,z,\theta) \le 0 \right] \right) \right\} \end{aligned} \tag{4.7}$$

where the weights ω^i correspond to discrete probabilities for the selected finite number of parameter points $\theta^i \in T$, i=1,2,....N. Since very often the probability distribution functions for the parameters are not available at the design stage, these weights may be selected by the designer so as to reflect subjective probabilities.

With the above simplification, the number of decision variables in (4.7) is finite, since optimization is performed over the vector d of design variables and the finite number of vectors $z^1,z^2,\ldots,z^N$ of control variables. The control variables z^i are selected to satisfy the corresponding constraints $f(d,z^i,\theta^i) \le 0$, so as to result in an optimal feasible operation at the point θ^i of the

parameter space. Note that despite this discretization, the feasibility constraint is still imposed so as to restrict the choice on the design d to guarantee feasible operation for every point $\theta \in T$. Thus, the formulation in (4.7) is a *nonlinear semi-infinite program* (NLSIP), with a finite number of decision variables and infinite number of constraints.

It is interesting to note that if the feasibility constraint is excluded in (4.7), the resulting structure of the problem is equivalent to that of a deterministic multiperiod problem, where the plant operates in each period with the parameter value θ^i, and with length of period proportional to ω^i. As discussed in Chapters 2 and 3, this class of multiperiod design problems can be solved very efficiently with the projection-restriction strategy. The question that immediately arises then is whether a finite number of points in θ-space can be selected so that by ensuring feasibility of the design for those points, one can guarantee that the feasibility constraint in (4.7) will be satisfied. If such a choice of finite number of parameter values were possible one could clearly solve problem (4.7) as an equivalent deterministic multiperiod design problem. In order to answer this question, it is essential first to reformulate the feasibility constraint in (4.7) so as to make it more amenable for analysis.

4.3 REFORMULATION OF THE FEASIBILITY CONSTRAINT

The logical constraint

$$\forall\ \theta \in T \left\{ \exists z \left(\forall\ j \in J \left[\ f_j(d,z,\theta) \le 0\ \right] \right) \right\} \tag{4.8}$$

which ensures overall feasibility of the design is the major source of computational difficulty in numerical solution of the problem represented by the NLSIP in (4.7). The reason is that it involves an infinite number of

inequality constraints for which feasibility has to be tested. The following theorem provides a possibility for circumventing this problem.

Theorem 1. The logical constraint (4.8) and the *max-min-max constraint,*

$$\max_{\theta \in T} \min_{z} \max_{j \in J} f_j(d,z,\theta) \leq 0 \tag{4.9}$$

are exactly equivalent.

Proof: This theorem can be proved in two parts, namely,

(4.8) $\Rightarrow$ (4.9) and (4.9) $\Rightarrow$ (4.8),

as given by Polak and Sangiovanni-Vincentelli (1979), and by Halemane and Grossmann (1981). An alternative proof which is simpler and more direct is given here. By the definition of the terms and relationships used in (4.8) and (4.9), the following equivalences apply, by considering *global max* and *min* operators:

$$\forall\ \theta \in T \left\{ \exists\ z \left(\forall\ j \in J \left[\ f_j(d,z,\theta) \leq 0\ \right]\right)\right\}$$

$$\Longleftrightarrow\ \forall\ \theta \in T \left\{ \exists\ z \left(\max_{j \in J} f_j(d,z,\theta) \leq 0 \right)\right\}$$

$$\Longleftrightarrow\ \forall\ \theta \in T \left\{ \min_{z} \max_{j \in J} f_j(d,z,\theta) \leq 0 \right\}$$

$$\Longleftrightarrow\ \max_{\theta \in T} \min_{z} \max_{j \in J} f_j(d,z,\theta) \leq 0$$

From these steps the equivalence of the first and last relations is established.

Q.E.D.

With this alternative and equivalent formulation of the feasibility constraint the optimal design problem in (4.7) can be rewritten as

$$\begin{aligned} &\underset{d,z^1,z^2,\ldots,z^N}{\text{minimize}} \quad \sum_{i=1}^{N} \omega^i \; C(d,z^i,\theta^i) \\ &\text{s.t.} \qquad f(d,z^i,\theta^i) \le 0, \quad i = 1,2,\ldots,N \qquad\qquad (4.10) \\ &\qquad\qquad \max_{\theta\in T} \; \min_{z} \; \max_{j\in J} \; f_j(d,z,\theta) \le 0 \end{aligned}$$

In order to describe qualitatively the significance of the *max-min-max* constraint, note that the inequality constraints are satisfied for non-positive values of the constraint functions $f_j(d,z,\theta)$, $j \in J$. Hence, the *worst constraint* function is that which is most likely (to be) violated, and is denoted by the index $\bar{j}$ which corresponds to the maximum valued function $f_{\bar{j}}(d,z,\theta)$ for given d, z, θ. The control $\bar{z}$ that minimizes this function $f_{\bar{j}}(d,z,\theta)$ for any given d, θ, corresponds to the *most feasible operation* (even) for the worst constraint. Then, a *critical parameter value* θ^c can be defined as that for which the worst constraint $f_{\bar{j}}$ is maximized while having the control $\bar{z}$, for a given design d. If for a design d, a control variable $\bar{z}$ can be chosen to satisfy the constraints at every critical parameter value θ^c, then the design d can be guaranteed to have feasible operation at every $\theta \in T$.

Note that in the above formulation the max-min-max constraint provides the possibility of circumventing the problem of simultaneously handling the infinite number of inequality constraints. The reason is that the max-min-max constraint determines a point θ^c for which the inequalities are most likely to be violated, while requiring that these inequalities be satisfied at that point. However, this constraint involves solving the *max-min-max problem,*

$$\max_{\theta\in T} \; \min_{z} \; \max_{j\in J} \; f_j(d,z,\theta) \qquad (4.11)$$

which is structurally far more complex than the min-max problems generally addressed in literature (Danskin, 1967; Demyanov and Malozemov, 1974). The main computational difficulty arises because of the non-differentiability of the max-min-max functions as well as the difficulties associated with evaluation of such complex functions (Polak and Sangiovanni-Vincentelli, 1979). Furthermore, it is not clear under which circumstances the solution of this subproblem is unique, since in Theorem 1 global max and min operators had to be used for the proof. Therfore, it is desirable as a next step to examine the properties and interpretation of the max-min-max constraint so as to gain a better insight and understanding from it.

4.4 INTERPRETATION AND PROPERTIES OF THE MAX-MIN-MAX CONSTRAINT

The max-min-max constraint can be written as a constrained max-min problem, by introducing an extra variable u,

$$\max_{\theta \in T} \; \min_{z} \left\{ u \;\middle|\; u \geq f_j(d,z,\theta), \;\; \forall\, j \in J \right\} \leq 0 \tag{4.12}$$

It then follows that for a given point $\theta \in T$ the value $\psi(d,\theta)$ determined by

$$\psi(d,\theta) \;=\; \min_{z} \left\{ u \;\middle|\; u \geq f_j(d,z,\theta), \;\; \forall\, j \in J \right\} \tag{4.13}$$

indicates the extent of (in)feasibility of operation of the design d for that point θ. A negative value of $\psi(d,\theta)$ indicates a finite region of feasibility and a positive value indicates infeasibility. Thus, the value $\psi(d,\theta)$ can be interpreted to be a good *measure of (in)feasibility* of operation at the chosen point $\theta \in T$. Since the constraint in (4.12) leads to a point θ^c which maximizes $\psi(d,\theta)$, θ^c corresponds to a *critical point* in the parameter space for which the

design d has either the smallest degree of feasibility (if $\psi(d,\theta^c) \leq 0$), or the largest degree of infeasibility (if $\psi(d,\theta^c) > 0$).

To illustrate these ideas consider the following set of two constraints which involve one variable z and one parameter θ:

$$\begin{aligned} f_1 &= -z + \theta && \leq 0 \\ f_2 &= z - 2\theta - d + 2 && \leq 0 \\ & 1 \leq \theta \leq 2 \end{aligned} \tag{4.14}$$

Figure 4.1a shows a plot of the feasible region on θ-z space, for a design corresponding to d = 0.5. As can be observed in the figure, the size of the feasible region increases as θ increases, with $\theta = 1$ being infeasible, $\theta = 1.5$ being feasible at one single value of z, and $\theta = 2$ having a finite region of feasibility. This can also be observed from Figure 4.1b where the feasible region on z-u space is shown for these three different θ-values. The value of ψ is determined by solving the problem,

$$\begin{aligned} \psi(d,\theta) &= \min_z \ u \\ \text{s.t.} \quad & f_1 = -z + \theta \leq u \\ & f_2 = z - 2\theta - d + 2 \leq u \\ & 1 \leq \theta \leq 2 \end{aligned} \tag{4.15}$$

for d = 0.5 and $1 \leq \theta \leq 2$, and its results are plotted in Figure 4.1c. Note that $\psi = 0$ for $\theta = 1.5$ which has a single point of feasibility. Also, negative values of ψ correspond to finite regions of feasibility as for instance at $\theta = 2$, and positive values of ψ are associated with infeasibility as for $\theta = 1.0$, which is a critical point where the maximum of ψ is attained. Note also that ψ decreases monotonically with increasing θ, while the feasible region gets expanded. From these observations it is clear that for a given

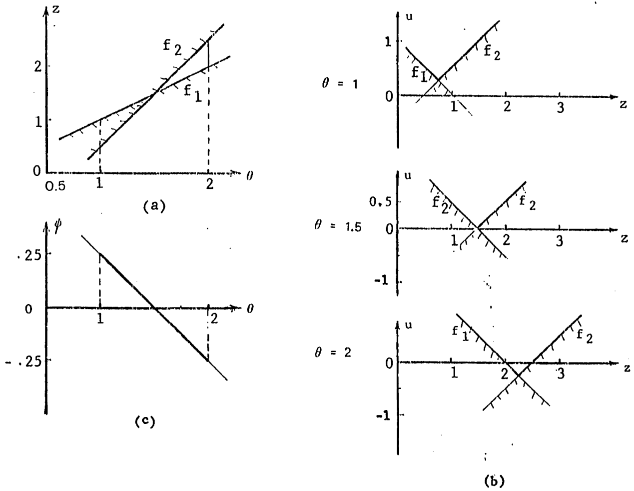

Figure 4.1: Feasible region and $\psi(d,\theta)$ for constraints (4.14) with d=0.5

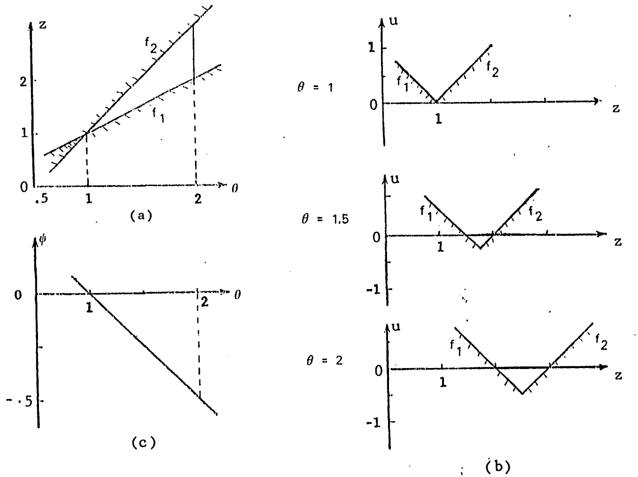

Figure 4.2: Feasible region and $\psi(d,\theta)$ for constraints (4.14) with d=1.0

design d and for any chosen parameter value θ, the value of $-\psi(d,\theta)$ can be interpreted as a measure of the *depth of the feasible region*, in the z-u space, for operation. This region corresponds to the projection of the actual overall feasible region in the d-θ-z-u space onto the z-u space for fixed values of d and θ.

To study the effect of changes in d, the region of feasibility for (4.14) is shown in Figure 4.2a and 4.2b for d = 1.0, and the corresponding plot of ψ is given in Figure 4.2c. By changing d from 0.5 to 1.0, overall feasibility has been achieved for all values of θ in the specified range $1 \leq \theta \leq 2$. Again, it is clear from Figure 4.2a and 4.2b that $\theta = 1$ is a critical point, since it corresponds to the maximum value of $\psi(d,\theta)$ for the specified range $1 \leq \theta \leq 2$. Thus, the design d = 1.0, which is feasible for the critical point $\theta = 1$ is found to be also feasible for the entire range $1 \leq \theta \leq 2$.

The example above would suggest that feasible operation in the design can be guaranteed by considering one single critical θ-point. The existence of a single critical parameter value can be associated with the monotonicity of $\psi(d,\theta)$ with respect to θ, which is in fact observed in this example (see the plot of ψ versus θ in Figure 4.2c). However, this may not be true in the general case, as is easily observed if a third constraint is considered together with the two others in (4.14) to give:

$$\begin{aligned} f_1 &= -z + \theta \leq 0 \\ f_2 &= z - 2\theta - d + 2 \leq 0 \\ f_3 &= -z + 6\theta - 9d \leq 0 \\ & 1 \leq \theta \leq 2 \end{aligned} \tag{4.15}$$

The feasible region for this set of constraints is shown in Figure 4.3a and Figure 4.3b for d = 1.0, and the corresponding function ψ is shown in Figure 4.3c. Note that $\psi(d,\theta)$ is no longer monotonic in θ, and is in fact non-

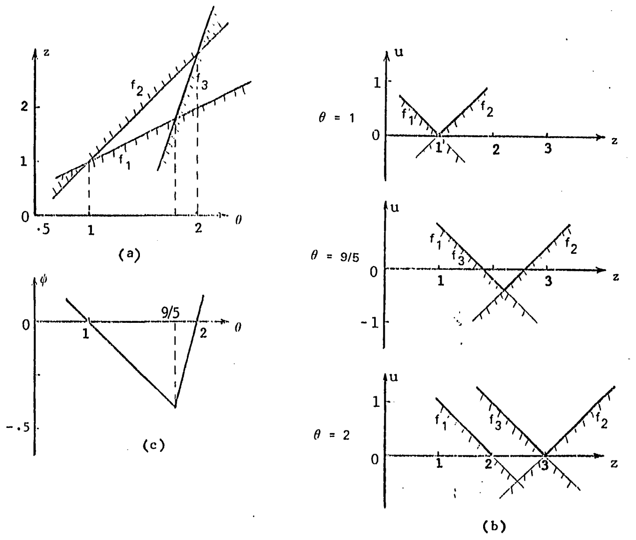

Figure 4.3: Feasible region and $\psi(d,\theta)$ for constraints (4.16) with d=1.0

differentiable at $\theta = 9/5$, and exhibits two local maxima at $\theta = 1$ and $\theta = 2$. It is clear from Figure 4.3a and Figure 4.3b that the size (depth) of the feasible region decreases at both extreme points, $\theta = 1$ and $\theta = 2$, and gets enlarged towards the interior point $\theta = 9/5$. There are in this case two critical points to be considered for design, which are in fact the two extreme points of the range $1 \le \theta \le 2$. This observation about the location of critical points can be generalized for the case of a set of nonlinear convex constraint functions through the following theorem:

Theorem 2. If the constraint functions $f_j(d,z,\theta)$ are jointly convex in z and θ, then the problem

$$\max_{\theta \in T} \min_{z} \max_{j \in J} f_j(d,z,\theta) \tag{4.17}$$

has its global solution θ^c at an extreme point of the polyhedral region $T = \left\{ \theta \mid \theta^L \le \theta \le \theta^U \right\}$.

Proof: This theorem can be proved in three parts as follows:

Property 1. If for every $j \in J$, $f_j(d,z,\theta)$ is jointly convex in z and θ, then $\phi(d,z,\theta) = \max_{j \in J} f_j(d,z,\theta)$ is also jointly convex in z and θ.

Proof: Since for any d, $f_j(d,z,\theta)$ is jointly convex in z and θ, the epigraph

$$\mathrm{Epi}\left\{ f_j(d,z,\theta) \right\} = \left\{ y,z,\theta \mid y \ge f_j(d,z,\theta) \right\}$$

is a convex set, for every $j \in J$.

Hence, $\bigcap_{j \in J} \mathrm{Epi}\left\{ f_j(d,z,\theta) \right\}$ is also a convex set (see e.g. Stoer and Witzgall, 1970; Rockefellar, 1970).

$$\text{But,} \quad \bigcap_{j \in J} \text{Epi}\left\{ f_j(d,z,\theta) \right\} = \left\{ y,z,\theta \mid y \geq f_j(d,z,\theta), \forall j \in J \right\}$$

$$= \left\{ y,z,\theta \mid y \geq \max_{j \in J} f_j(d,z,\theta) \right\}$$

$$= \text{Epi}\left\{ \max_{j \in J} f_j(d,z,\theta) \right\}$$

$$= \text{Epi}\left\{ \phi(d,z,\theta) \right\}$$

which is therefore a convex set. From this it follows that the function $\phi(d,z,\theta)$ is also jointly convex in z and θ.

Property 2. If $\phi(d,z,\theta)$ is jointly convex in z and θ,

then $\psi(d,\theta) = \min_z \phi(d,z,\theta)$ is convex in θ.

Proof: Let $\psi(d,\theta^3) = \min_z \phi(d,z,\theta^3) = \phi(d,z^3,\theta^3)$.

Let θ^1, $\theta^2 \in T$ be two distinct points that are different from θ^3, and $0 < \lambda < 1$, such that $\theta^3 = (1-\lambda)\,\theta^1 + \lambda\,\theta^2$.

Let $\psi(d,\theta^1) = \min_z \phi(d,z,\theta^1) = \phi(d,z^1,\theta^1)$,

and $\psi(d,\theta^2) = \min_z \phi(d,z,\theta^2) = \phi(d,z^2,\theta^2)$.

Since ~~$\phi(d,z,\theta)$ is jointly convex in z and θ,~~

$$\left(1-\lambda\right) \phi(d,z^1,\theta^1) + \lambda\ \phi(d,z^2,\theta^2) \geq \phi(d,z^{12},\theta^3),$$

where $z^{12} = (1-\lambda)\, z^1 + \lambda z^2$.

But, $\phi(d,z^{12},\theta^3) \geq \min_z \phi(d,z,\theta^3) = \phi(d,z^3,\theta^3)$,

and therefore, $\left(1-\lambda\right) \phi(d,z^1,\theta^1) + \lambda\ \phi(d,z^2,\theta^2) \geq \phi(d,z^3,\theta^3)$,

or, $$\left(1-\lambda\right) \psi(d,\theta^1) + \lambda\ \psi(d,\theta^2) \geq \psi(d,\theta^3).$$

Noting that $\theta^3 = (1-\lambda)\ \theta^1 + \lambda\ \theta^2$, it is clear that $\psi(d,\theta)$ is convex in θ.

Property 3. If $\psi(d,\theta)$ is convex in θ,

then *every local solution* θ^o of the problem

$$\max_{\theta \in T} \psi(d,\theta)$$

lies at an extreme point of the convex region T, unless the solution is degenerate.

Proof: Assume that θ^o is a non-extreme point of the region T.

Then it is possible to choose two distinct points θ^1, $\theta^2 \in$ T in the neighborhood of θ^o, and $0 < \lambda < 1$ such that $\theta^o = (1-\lambda)\ \theta^1 + \lambda\ \theta^2$.
Since $\psi(d,\theta)$ is convex in θ,

$$\left(1-\lambda\right) \psi(d,\theta^1) + \lambda\ \psi(d,\theta^2) \geq \psi(d,\theta^o).$$

Since θ^o is a non-degenerate solution for the above maximization problem,

$$\max_{\theta \in T} \psi(d,\theta) > \left(1-\lambda\right) \psi(d,\theta^1) + \lambda\ \psi(d,\theta^2)$$

Therefore, $$\max_{\theta \in T} \psi(d,\theta) > \psi(d,\theta^o),$$

which is a contradiction since θ^o maximizes locally the function $\phi(d,\theta)$. Hence, the assumption that θ^o is a non-extreme point of the region T must be incorrect, and this proves the result stated above.

Property 4. If the region T is polyhedron defined by (4.1),

the *global solution* θ^c for the problem (4.17)

must lie at a *corner point (vertex)* of this polyhedron,

unless the solution is degenerate.

This result is obvious from the fact that the vertices are the only extreme points for a polyhedral region, and that every boundary point (as well as interior point) can be expressed as a convex combination of these extreme points (vertices). Therefore, any local solution to the problem (4.18), and hence its global solution θ^c, must lie at a vertex of the polyhedral region T. Thus the result stated in Theorem 2 is proved.

4.5 DISCUSSION

Since there are a finite number of vertices for the polyhedron T, Theorem 2 provides an answer to the question as to whether a finite number of points can be considered for design to ensure feasibility for all the points in the polyhedron T. It follows from Theorem 2 that if the constraints are convex, feasibility of operation for every value of $\theta \in T$ can be guaranteed by considering in the design all the vertices of the polyhedron T, since all the critical parameter values θ^c will then be accounted for. Also, since (4.17) represents a maximization of a convex function as shown in Property 3 of Theorem 2, there can be multiplicity of local solutions for (4.17), and hence a number of different critical points. This result indicates that in optimizing a design, it is important to ensure feasibility for every one of the possibly several critical points. The solution algorithms presented in the next chapter provide a systematic procedure to ensure feasibility of the design for every critical parameter point by explicitly considering them for design.

It should also be clear that the assumption of convexity on the

constraint functions in Theorem 2 is a sufficient condition for the location of the critical points at the vertices of the polyhedron T. Therefore, there can also be cases where even if non-convex constraint functions are involved, the critical points correspond to vertices. However, it is clear that this will not always necessarily be true. Despite this limitation, the result provided by Theorem 2 is very useful for deriving efficient algorithms for solving the problem of design under uncertainty, as shown in the next chapter.

CHAPTER 5

SOLUTION ALGORITHMS FOR DESIGN UNDER UNCERTAINTY

As shown in Chapter 4, when solving the problem of optimal design under uncertainty, it is essential to satisfy the max-min-max constraint in order to guarantee feasibility of operation for every $\theta \in T$. This implies that if for a given design d, feasibility holds for every critical parameter value θ^c, then feasibility can be guaranteed for every $\theta \in T$. In this chapter, two solution algorithms are presented which assume that the critical points are located at the vertices of the polyhedral region T of the parameters. These algorithms transform the problem of design under uncertainty into an iterative multiperiod design problem, for which an efficient decomposition strategy has already been presented in Chapter 2. Finally, a brief discussion is presented on the problems and limitations of locating critical parameter points.

5.1 ALGORITHM 1.

As was proved in Theorem 2, if the inequality constraint functions f_j, $j \in J$ are convex, then the critical points must lie at the vertices of the polyhedron T. Since there are a finite number of vertices in T, a design obtained by considering all these vertices will be feasible for every point in the polyhedron. This would then suggest the following algorithm for solving the design problem:

Step 1 - Include all the N vertices in the set

$$T_o = \left\{ \theta^i \mid \theta^i \text{ is a vertex of T, } i=1,2,\ldots N \right\} \tag{5.1}$$

Step 2 - Solve the problem

$$\begin{aligned} \underset{d,\, z^1, z^2, \ldots z^N}{\text{minimize}} \quad & C^o(d) + \sum_{i=1}^{N} \omega^i \, C(d, z^i, \theta^i) \\ \text{s.t.} \quad & f(d, z^i, \theta^i) \le 0 \quad i = 1,2,\ldots N \\ & \theta^i \in T_o \end{aligned} \tag{5.2}$$

with the *projection-restriction strategy*, so as to obtain the design d^o.

Since T_o includes all the N vertices of the polyhedron T, every critical point θ^c corresponding to the above design d^o will also be included in T_o. Therefore the design d^o will be feasible for its critical points, and hence it will be feasible for every $\theta \in T$.

The drawback in this algorithm is that the number of vertices N to be considered for design increases exponentially with the number of parameters p, since $N = 2^p$. Thus, for a problem involving ten uncertain parameters (p = 10), the design problem has to consider 2^{10} = 1024 vertices, which would lead to

an extremely large problem in (5.2). Because of this fact, Algorithm 1 is only suitable when the number of parameters is small (typically, not more than five).

5.2 ALGORITHM 2.

When the number of uncertain parameters is large, it is desirable to have an efficient solution procedure which does not have to consider all the vertices explicitly as in Algorithm 1, but that can still ensure the optimality and feasibility of the design. The following algorithm is proposed for this purpose, wherein only a subset of the vertices are considered for design in the resulting multiperiod formulation.

Step 1 - Set k = 0.

Choose an initial set T_o consisting of N_o vertices where $N_o \ll 2^p$.

This can be achieved with small computing requirements, using the procedure suggested by Grossmann and Sargent (1978), in which each constraint is maximized individually by assuming monotonicity and analyzing the signs of their gradients. The gradients $\partial f_j/\partial \theta_k$ of each of the individual constraint functions f_j, j=1,2,....m, with respect to the parameters θ_k ,k=1,2,....p, are computed at initial values of d and z, and the signs of these gradients are analyzed. If for each individual constraint function f_j, the gradient $\partial f_j/\partial \theta_k > 0$, the upper bound θ_k^U is selected for the parameter θ_k, whereas if $\partial f_j/\partial \theta_k < 0$ the lower bound θ_k^L is selected. Clearly, for zero gradients either choice of the bounds is possible. Since each constraint may lead to a different vertex, the set of vertices obtained for all constraints is finally merged into into the smaller set of vertices T_o by using a set covering formulation. It should be noted that *if the constraint functions* f_j, *are*

monotonic in the parameters θ_k, these vertices will correspond to the maximization of individual constraint functions (see Grossmann, 1977).

Step 2 - Solve the problem

$$\underset{d,\ z^i, i=1,2,\ldots N^k}{\text{minimize}} \quad C^o(d) + \sum_{i=1}^{N_k} \omega^i \ C(d,z^i,\theta^i) \tag{5.3}$$

$$\text{s.t.} \qquad f(d,z^i,\theta^i) \ \leq \ 0, \quad i = 1,2,\ldots N^k$$

so as to obtain the design d^k.

Step 3 - Determine the critical parameter values $\theta^{c,k}$ by solving for every vertex θ^i in T_k, the problem

$$\psi(d^k,\theta^i) \ = \ \min_z \left\{ \ u \ \middle| \ u \geq f_j(d^k,z,\theta^i), \ j \ \in J \right\} \tag{5.4}$$

The vertex that gives rise to the maximum value of ψ is then determined and is denoted by $\theta^{c,k}$. If $\psi(d^k,\theta^{c,k}) \leq 0$, stop. Otherwise, proceed to Step 4.

Step 4 - Define $\quad T_{k+1} \ = \ T_k \bigcup \left\{ \ \theta^{c,k} \ \right\}, \quad N_{k+1} = \mid T_{k+1} \mid, \quad$ (5.5) set $k = k+1$ and iterate from Step 2.

Note that at the termination of this algorithm the design will necessarily be feasible for all values of parameters, because it will be feasible for the critical parameters values. Also, the algorithm has to terminate in a finite number of iterations since there are only a finite number of critical parameter points to be considered. The initial vertices predicted in Step 1 by the method of Grossmann and Sargent (1978) will often yield very good guesses for which only one iteration in Algorithm 2 may be required. As in

Algorithm 1, problem (5.3) in Step 2 can be solved with the projection-restriction strategy described in Chapter 2. Also, note that the minimizations in (5.4) may not have to be performed until completion for all vertices, as they can be stopped when ψ reaches a negative value in which case the existence of a non-empty feasible region is detected. Thus, by the above considerations Algorithm 2 will provide in general a much more efficient method of solution than Algorithm 1.

However, there are two factors in Algorithm 2 which would require further investigation. One of them is the number of parameter points considered for design, which in turn determines the size of problem (5.3) in Step 2. This number will increase at each iteration since a new parameter point will be added. The question is whether this number can be kept small throughout, by eliminating some of the previous points while adding new ones. This elimination can probably be performed on the basis of the value of ψ, to remove those vertices corresponding to the minimum value of ψ. From the discussions in Chapter 4 it is clear that a minimum value of ψ indicates a feasible region with the largest *depth* and hence most likely to remain feasible for small changes in the design d. It should be noted that no theoretical justification is given for the elimination of vertices in this manner, and it is to be considered only as a very good heuristic. A second question is whether it is possible to determine the critical parameter points in Step 3 without explicitly analyzing each of the individual vertices and solving for $\psi(d,\theta)$. If this were possible, it would further enhance the efficiency of the above solution strategy when dealing with a large number of parameters.

5.3 NUMERICAL EXAMPLES

To illustrate the application of the algorithm described above, two example problems are presented below.

Example 1.

In this example the heat exchanger network 4SP1 of Lee et al. (1970) with outlet temperatures specified as inequalities is considered (see Grossmann and Sargent, 1978). The flowsheet consists of five heat exchangers, one of which is a steam heater and another being a cooler using cooling water, and with two hot streams and two cold streams as shown in Figure 5.1. Table 5.1 gives the data for the problem. The overall heat transfer coefficients U_i, $i=1,2,...5$, were considered to be the parameters with ±20% uncertainty in their values. The design problem then consists in selecting the areas A_i, $i=1,2,....5$ so that irrespective of the actual values of the heat transfer coefficients (within the ±20% range around their nominal values), the specifications on the outlet temperatures are satisfied by suitable choice of the cooling water outlet temperature T_{15} and the steam temperature T_{13}. Apart from the equality constraints representing the heat balance and design equations for the network, the following inequality constraints on the temperatures of various streams have to be satisfied.

$$T_3 \geq 534$$

$$T_6 \geq 434$$

$$T_{10} \geq 411$$

$$T_{12} \geq 367$$

$$T_5 - T_8 \leq 0$$

$$T_3 - T_{13} \geq 0.55$$

$$T_{11} - T_5 \geq 0.55$$

$$T_7 - T_2 \geq 0.55$$

$$T_8 - T_6 \geq 0.55$$

$$T_9 - T_{15} \geq 0.55$$

Among the above constraints, the first four constraints correspond to specifications on the outlet temperatures, and the last five on the minimum temperature approach. Table 5.2 gives the initial set of vertices considered for design, which were obtained by analyzing the signs of gradients of individual constraints as suggested by Grossmann and Sargent (1978). Note that this set includes also the nominal point so as to provide an adequate weighting in the cost function. The CPU-time required to solve this multiperiod design problem with the projection-restriction strategy was only 16.7sec (DEC-20). The design corresponding to these five parameter points was found to be feasible for all the 32 vertices. The results are given in Table 5.3, from which it is clear that the values of ψ are non-positive at all the vertices, thus ensuring feasibility.

Note that the actual value of ψ is dependent on the scaling factors used for the constraint functions, which are given in Table 5.1 for this problem. Although the choice of these scaling factors is arbitrary, it does not affect the detection of (in)feasibility for a given design. The CPU-time to determine ψ for the 32 vertices and thus test for feasibility was only 18.0sec (DEC-20). It is also interesting to note that there were many vertices with the same value of $\psi(d,\theta)$, which may be attributed to the fact that the constraints themselves are sparse with respect to these uncertain parameters. In solving this example problem, optimization was performed in each case with the variable metric projection algorithm of Sargent and Murtagh (1973). In this example the optimum design solution was obtained with a single iteration through

Algorithm 2, without the need for considering additional vertex points. This may not always be the case as can be seen from the second example which follows.

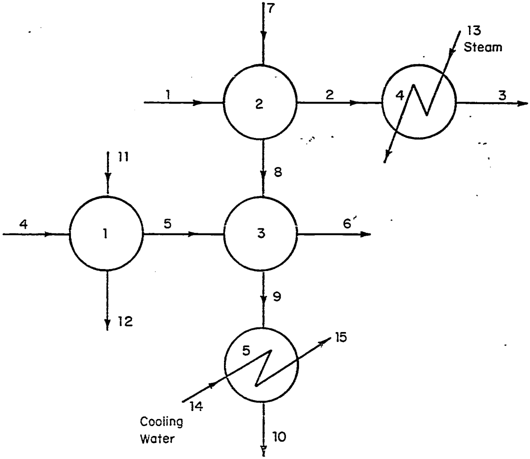

Figure 5.1: Heat exchanger network for Example 1.

Table 5.1: Data for Example 1.

Nominal values of the uncertain parameters:

$U_1 = U_2 = U_3 = U_5 = 3066$ kJ / m^2 hr K

$U_4 = 4088$ kJ / m^2 hr K

Other process parameters:

$FC_1 = 21912$ kJ / hr K	$T_1 = 389$ K
$FC_4 = 27461$ kJ / hr K	$T_4 = 333$ K
$FC_7 = 38009$ kJ / hr K	$T_7 = 522$ K
$FC_{11} = 31674$ kJ / hr K	$T_{11} = 433$ K
	$T_{14} = 311$ K

Cost function:

$$C = 145.6 \sum_{i=1}^{5} A_i^{0.6} + \sum_{i=1}^{5} \omega^i \left(18.5\, F_s^i + 0.923\, F_w^i\right)$$

$\omega^1 = 0.6, \quad \omega^i = 0.1, \quad i=2,3,4,5.$

Scaling factors for constraints: 1.8

Bounds on control variables:

$314 \leq T_{15} \leq 355$ K

$534 \leq T_{13} \leq 556$ K

Table 5.2: Parameter values considered for design in Example 1.

	U_1	U_2	U_3	U_4	U_5
1	N	N	N	N	N
2	U	U	U	U	N
3	L	U	U	U	N
4	U	L	L	L	L
5	L	U	L	U	N

N - Nominal, L - Lower bound U - Upper bound

Table 5.3: Results for Example 1.

(a) Heat exchange areas, m^2

A_1	A_2	A_3	A_4	A_5	cost, $/yr
30.8	62.2	45.58	3.9	2.9	11758

CPU-Time (DEC-20) for obtaining the design

using the projection-restriction strategy: 16.7sec

(b) Test for feasibility at the vertices:

$\psi(d,\theta)$	Number of vertices	Vertex number, v
0.0	24	0-15, 24-31
-0.679	Nominal Point	
-2.452	4	20-23
-3.775	4	16-19

CPU-Time (DEC-20) for testing feasibility

and determining $\psi(d,\theta)$ at the vertices: 18.0sec

Vertex number v given by:

$$v = \sum_{i=1}^{5} \sigma_i (2)^{5-i}, \qquad \sigma = \begin{cases} 0, & \text{if } U_i = U_i^L \\ 1, & \text{if } U_i = U_i^U \end{cases}$$

Example 2.

Figure 5.2 shows the flowsheet consisting of a reactor and a heat exchanger, used to handle a first-order exothermic reaction A → B, for which the problem data is given in Table 5.4. The parameters considered to have uncertainty in their values are: (i) F_o, the feed flow rate (±10%); (ii) T_o, the temperature of the feed stream (±2%); (iii) T_{w1}, the inlet temperature of cooling water (±3%); (iv) k_R, the Arrhenius rate constant (±10%); and (v) U, the overall heat transfer coefficient for the cooler(±10%). Among these five parameters, the first three are associated with inlet streams to the units while the latter two correspond to internal parameters of the process. The conversion is specified to be not less than 90%, and the temperature of the reactor must be lower than the specified upper bound, 389°K. The design problem consists in selecting the optimal sizes for the reactor and heat exchanger so that the specifications can be satisfied by suitable choice of the temperatures T_1, T_2, T_{w2}, in Figure 5.2, irrespective of the actual values of the parameters. The material and heat balance equations and design equations for the reactor and heat exchanger represent the equality constraints of the design problem, and are similar to the equations (3.9)-(3.14). Other specifications to be satisfied are expressed by the following inequality constraints:

(a) $$\hat{V} \geq V$$

(b) $$\left(C_{Ao} - C_{A1}\right) / C_{Ao} \geq 0.90$$

(c) $$333 \leq 389$$

(d) $$T_1 - T_2 \geq 0$$

(e) $$T_{w2} - T_{w1} \geq 0$$

(f) $$T_1 - T_{w2} \geq 11.1$$

(g) $$T_2 - T_{w1} \geq 11.1$$

The initial set of parameter points consists of the nominal point and three vertices, obtained by analyzing the gradient of the constraints, as given in Table 5.5a. The design corresponding to these four points is given in Table 5.6a. This design is found to be infeasible for eight of the thirty-two vertices as indicated in Table 5.6b by the positive values of ψ for these eight vertices. Since the value of ψ is found to be the same for all these eight vertices, one among them is chosen to be added to the initial set of vertex points considered for design. This new set of parameter points is given in Table 5.5b and the resulting design shown in Table 5.7a. This design is feasible for all the 32 vertices as shown by the non-positive values of ψ given in Table 5.7b. Here again, these values of ψ correspond to the scaling factors given in Table 5.4 for the constraints of the problem. This example illustrates the need to analyze the max-min-max constraint as a means to achieve feasibility of operation for the specified range of parameter values. Again, as in the previous example, there are many vertices having the same value of $\psi(d,\theta)$, which can be attributed to the fact that the constraints are sparse with respect to the parameters. Note that it took 9.2sec and 12.8sec to solve the multiperiod design problem with 4 and 5 periods respectively, in Step 2 of Algorithm 2. As can be seen from Tables 5.6 and 5.7, the computational time taken to test for feasibility was comparatively large - 65.8sec and 73.9sec respectively in the two iterations through Algorithm 2. It took a total of 161.7 sec (DEC-20) for obtaining the complete solution, which is quite moderate, considering the size and complexity of this problem. The optimizations were performed using the variable metric projection algorithm of Sargent and Murtagh (1973).

Finally, the sensitivity of the solution with respect to different choices of weighting factors (in the objective function) was tested for this problem. Table 5.8 gives the results obtained for this design problem by taking different weighting factors ω^i in the objective function. As can be seen from this table, the volume of the reactor is not dependent on the weighting factors, whereas the heat transfer area and the annual cost varies within ±11.5% and ±5.7% respectively. It is to be noted that the design does not get considerably affected for moderate changes in the weighting factors. It is only when $\omega^0 \rightarrow 1.0$ that the variations seem to be quite substantial, since in this case most of the weighting is given to the nominal point. Although the actual design solution depends on these weighting factors, the feasibility of the design is always guaranteed. This is because the constraints corresponding to the critical parameter points are satisfied at the solution, and it is only through the sensitivity of the objective function that the weighting factors affect the optimum design solution. However, this sensitivity is usually not large since the design has been optimized for several parameter points, unlike the common practice which is to optimize for a single parameter value.

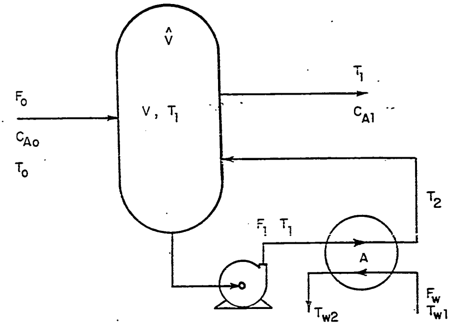

Figure 5.2: Reactor - Heat Exchanger System for Example 2.

Table 5.4: Data for Example 2

Nominal values of uncertain parameters:

k_R	= 0.6242	hr^{-1}	U	= 1635	$kJ/m^2\ hr\ K$
F_o	= 45.36	kgmole/hr	T_o	= 333	K
T_{w1}	= 300	K			

Other process parameters:

E/R	= 555.6	K	$-\Delta H_{rxn}$	= 23260	kJ/kgmole
C_{Ao}	= 32.04	$kgmole/m^3$	C_p	= 167.4	kJ/kgmole

Cost function:

$$C = 691.2\ \hat{V}^{0.7} + 873.6\ A^{0.6} + \sum_{i=1}^{N} \omega^i \left(1.76\ F_w^i + 7.056\ F_1^i \right)$$

$\omega^1 = \omega^0$; $\omega^i = (1 - \omega^0)/(N-1)$, i=2,3,....N

$\omega^0 = 0.5$, (weighting factor for the nominal point)

Scaling factors for constraints:

(a) 3.531; (b) 100.0; (c) 1.8; (d),(e),(f),(g) 18.0

Bounds on control variables	Starting Point			
	$\hat{V}$	=	7.09	m^3
	A	=	11.15	m^2
$311 \le T_1 \le 389$ K	T_1	=	367	K
$311 \le T_2 \le 389$ K	T_2	=	333	K
$301 \le T_{w2} \le 355$ K	T_{w2}	=	350	K

Table 5.5: Parameter values considered for design in Example 2.

	k_R	U	F_o	T_o	T_{w1}
1	N	N	N	N	N
2	L	U	L	L	L
3	U	L	U	U	U
4	U	L	U	U	L
5	L	L	U	U	U

(a) Initial set of points: (1), (2), (3), (4)
(b) Second set of points: (1), (2), (3), (4), (5)

N - Nominal, L - Lower bound, U - Upper bound

Table 5.6: Results for Example 2, First iteration with Algorithm 2.

(a) Design obtained for parameter points given in Table 5.5a

$\hat{V} = 5.3\ m^3$, $A = 10.5\ m^2$ Cost: 10820 \$/yr

CPU-Time (DEC-20) for obtaining the design

using the projection-restriction strategy: 9.2sec

(b) Test for feasibility at the vertices:

$\psi(d,\theta)$	Number of vertices	Vertex number, v
+1.280	8	4-7, 12-15
0.0	Nominal Point	
0.0	16	0-3, 8-11, 20-23, 28-31
-1.151	8	16-19, 24-27

CPU-Time (DEC-20) for testing feasibility

and determining $\psi(d,\theta)$ at the vertices: 65.8sec

Vertex number v given by:

$$v = \sum_{i=1}^{5} \sigma_i (2)^{5-i}, \qquad \sigma = \begin{cases} 0, & \text{if } \theta_i = \theta_i^L \\ 1, & \text{if } \theta_i = \theta_i^U \end{cases}$$

$\theta_1 \equiv k_R$, $\theta_2 \equiv U$, $\theta_3 \equiv F_o$, $\theta_4 \equiv T_o$, $\theta_5 \equiv T_{w1}$

Table 5.7: Results for Example 2, Second iteration with Algorithm 2.

(a) Design obtained for parameter points given in Table 5.5a

$\hat{V} = 6.5\ m^3$, $A = 9.2\ m^2$ Cost: 10110 \$/yr

CPU-Time (DEC-20) for obtaining the design using the projection-restriction strategy: 12.8sec

(b) Test for feasibility at the vertices:

$\psi(d,\theta)$	Number of vertices	Vertex number, v
0.0	8	4-7, 12-15
-1.220	Nominal Point	
-1.220	16	0-3, 8-11, 20-23, 28-31
-2.323	8	16-19, 24-27

CPU-Time (DEC-20) for testing feasibility and determining $\psi(d,\theta)$ at the vertices: 73.9sec

---

Vertex number v given by:

$$v = \sum_{i=1}^{5} \sigma_i (2)^{5-i}, \qquad \sigma = \begin{cases} 0, & \text{if } \theta_i = \theta_i^L \\ 1, & \text{if } \theta_i = \theta_i^U \end{cases}$$

$\theta_1 \equiv k_R,\ \theta_2 \equiv U,\ \theta_3 \equiv F_o,\ \theta_4 \equiv T_o,\ \theta_5 \equiv T_{w1}$

Table 5.8: Results of Example 2, for different choices of weighting factors

ω^0	$\hat{V}$ m^3	A m^2	Annual Cost, $/yr
0.0	6.5	10.22	11396
0.1	6.5	10.13	11302
0.2	6.5	10.02	11206
0.3	6.5	9.91	11108
0.4	6.5	9.79	11007
0.5	6.5	9.66	10902
0.6	6.5	9.51	10793
0.7	6.5	9.35	10678
0.8	6.5	9.14	10553
0.9	6.5	8.86	10409
1.0	6.5	8.11	10160

Weighting factors for the vertices:
$\omega^1 = \omega^0$; (Nominal Point)

$\omega^i = (1 - \omega^0)/(N-1)$, i=2,3,....N

Number of periods considered N = 5.

5.4 DISCUSSION ON LOCATING THE CRITICAL PARAMETER POINTS

It is interesting to see that in the first example the design solution is obtained by considering only those vertices that were indicated by analyzing the gradients of individual constraint functions. As pointed out earlier in Section 5.2, *if the constraint functions are monotonic with respect to the parameters,* this procedure corresponds to the *maximization of the individual constraint functions* (see Grossmann and Sargent, 1978). However, it is clearly evident from Example 2 above that such procedure cannot always lead to the right set of parameters points required to be considered for design. The only method of characterizing these parameter points is by analyzing the max-min-max constraint. Thus, Example 2 illustrates this need for analyzing the max-min-max constraint as a means to achieve feasibility of operation for the specified range of parameter values. To understand this aspect in the overall design strategy, consider the following set of three linear constraints:

$$\begin{aligned} f_1 &= -z_1 + 3\theta_1 - \theta_2 \le 0 \\ f_2 &= -z_2 - \theta_1 + 3\theta_2 \le 0 \\ f_3 &= z_1 + z_2 - \theta_1 - \theta_2 - d \le 0 \\ &1 \le \theta_i \le 2, \quad i=1,2. \end{aligned} \tag{5.6}$$

Since these constraint functions are linear, their gradients are independent of the initial point chosen for such calculations; and since they are monotonic in θ, the maximization of each of these functions can be performed by analyzing these gradients. It is clear that the three constraint functions get maximized at the three different vertices $\theta^1 = [2,1]$, $\theta^2 = [1,2]$ and $\theta^3 = [1,1]$ respectively. Therefore, according to the procedure which is based solely on maximization of individual constraint functions, these three vertices would form the set of parameter points to be considered for design. The fact that this procedure can lead to designs that may be infeasible is shown below by

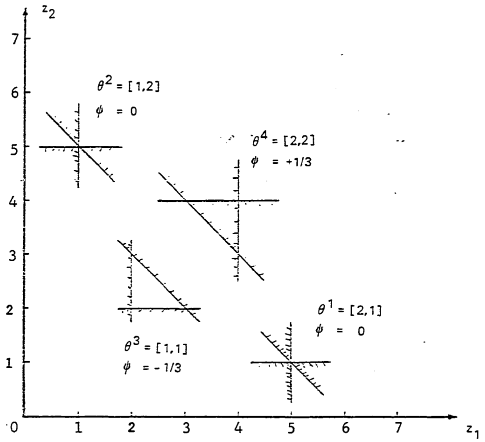

Figure 5.3: Feasible Region for the constraints in (5.6) for d = 3.

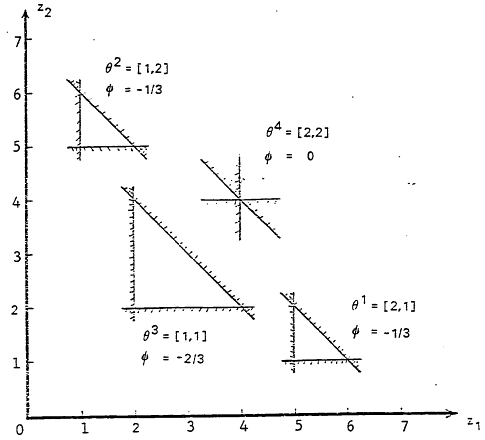

Figure 5.4: Feasible Region for the constraints in (5.6) for d = 4.

analyzing the max-min-max constraint for the above set of design constraints, which determines the value of ψ - the measure of (in)feasibility for any given design d. For a design d, the value of $\psi(d,\theta)$ for any parameter value θ is given by

$$
\begin{aligned}
\psi(d,\theta) \;=\; & \min_{z} \; u \\
\text{s.t.} \quad & f_1 = -z_1 + 3\theta_1 - \theta_2 \le u \\
& f_2 = -z_2 - \theta_1 + 3\theta_2 \le u \\
& f_3 = z_1 + z_2 - \theta_1 - \theta_2 - d \le u
\end{aligned}
\tag{5.7}
$$

Figure 5.3 gives a plot of the feasible region for the set of constraints in (5.6), with a design d = 3, wherein the values of ψ obtained from (5.7) are also given for each of the four vertex points of the parameter-space. Since the constraint functions are linear and hence convex, the critical parameter points must lie at a vertex, as shown in Theorem 4.2. The values of ψ in Figure 5.3 therefore indicate that $\theta^4 = [2,2]$ is the critical point, since ψ attains its maximum at this vertex. Also note in Figure 5.3 that the design d = 3 is found to be feasible for the three vertices $\theta^1 = [2,1]$, $\theta^2 = [1,2]$ and $\theta^3 = [1,1]$, whereas it is infeasible for the critical parameter point - namely the vertex $\theta^4 = [2,2]$. To make the design feasible for the critical point as well, consider d = 4. The feasible region and the value of ψ for the four vertices are again shown in Figure 5.4 for this value of d = 4. It is clear from Figure 5.4 that the design is just feasible for the critical point, whereas a finite region of feasibility exists for all other points in the parameter space. Thus, by ensuring the feasibility of the design for the critical parameter values, it is possible to guarantee the overall feasibility of the design for every parameter value within the specified range.

In order to gain some further insight as to why in some cases the maximization of individual constraint functions will lead to the correct critical

points, assume that for all $\theta \in T$ the same common (single) set of values $\bar{z}$ is selected for the control variables. It then follows that the max-min-max constraint reduces to

$$\max_{\theta \in T} \max_{j \in J} f_j(d,\bar{z},\theta) \leq 0$$

which is equivalent to (5.8)

$$\max_{\theta^j \in T} f_j(d,\bar{z},\theta^j) \leq 0, \quad \forall\, j \in J.$$

Thus if for the design design d, it is possible to select a control $\bar{z}$, feasible and common for all $\theta \in T$, then some of the parameter points predicted by maximization of individual constraints will correspond to those (critical points) predicted by the max-min-max constraint. However, it is clear that in general different controls z may have to be selected for different realizations of θ to maintain feasibility. Therefore, maximization of the individual constraint functions cannot always lead to the critical points, although it usually provides a very good guess for an initial set of parameter points to be considered for design.

Another interesting question about locating the critical points is to determine the conditions under which only *a single critical point* would suffice to be considered for design. As pointed out earlier in Chapter 4, monotonicity of $\psi(d,\theta)$ in θ gives rise to a single critical point in the space of parameters, as is also the case in the example given above in (5.6). Although the property of monotonicity of ψ is difficult to establish when the constraint functions are nonlinear, the following analysis can be applied for the case of linear constraint functions.

Suppose that for a given design d, the set of *linear constraint functions* are expressed as:

$$f_j = \sum_{k=1}^{p} a_{jk}\,\theta_k + \sum_{k=1}^{n_z} b_{jk}\,z_k + c_j \le 0, \quad j=1,2,....m \tag{5.9}$$

The critical point θ^c is given by the solution to the problem:

$$\max_{\theta \in T} \; \min_{z} u \tag{5.10}$$

$$\text{s.t.} \quad f_j = \sum_{k=1}^{p} a_{jk}\,\theta_k + \sum_{k=1}^{n_z} b_{jk}\,z_k + c_j \le u, \quad j=1,2,....m$$

Assume that for any $\theta \in T$ the problem

$$\psi(d,\theta) = \min_{z} u \tag{5.11}$$

$$\text{s.t.} \quad f_j = \sum_{k=1}^{p} a_{jk}\,\theta_k + \sum_{k=1}^{n_z} b_{jk}\,z_k + c_j \le u, \quad j=1,2,....m$$

has the *same set of active constraints* (e.g. the first r, $r \le m$).

The Kuhn-Tucker conditions for the above minimization problem then gives

$$\begin{aligned}
&\text{(a)} \quad 1 = \sum_{j=1}^{r} \lambda_j \\
&\text{(b)} \quad 0 = \sum_{j=1}^{r} \sum_{k=1}^{n_z} \lambda_j\, b_{jk}, \quad k = 1,2,....n_z \\
&\text{(c)} \quad u = \sum_{k=1}^{p} a_{jk}\,\theta_k + \sum_{k=1}^{n_z} b_{jk}\,\hat{z}_k + c_j, \quad j=1,2,....m
\end{aligned} \tag{5.12}$$

Since the value of u at the minimum determines ψ, it follows that

$$\psi = \frac{1}{r} \sum_{i=1}^{r} \left(\sum_{k=1}^{p} a_{jk}\, \theta_k + \sum_{k=1}^{n_z} b_{jk}\, \hat{z}_k + c_j \right) \tag{5.13}$$

From this expression it is clear that $\psi(d,\theta)$ is monotonic in θ, for the chosen design d. Note that in (5.12) the active constraints are determined by the values of the multipliers λ_j, which are in turn obtained from the subset (a),(b) in (5.12). Since there are r multipliers λ_j, and $n_z + 1$ equations in (5.12a,b), and since $r \leq m$, a necessary condition for having the same set of constraints to be active for every $\theta \in T$ would be: $m \leq n_z + 1$. In general, a necessary and sufficient condition would be to have the values of all the multipliers λ_j (j=1,2,....m) uniquely determined by the system (5.12). Furthermore, if for every k, $\mathrm{sign}(\partial\psi/\partial\theta_k) = \mathrm{sign}(a_{jk})$ for some constraints j, then these constraints are maximized at the point defined by max $\psi(d,\theta)$, and under these conditions the maximization of individual constraints with respect to θ would lead to the critical points.

To illustrate these ideas consider the set of constraints given by (5.6). The solution to (5.12 a,b) yields $\lambda_1 = \lambda_2 = \lambda_3 = +1/3$, indicating that the three constraints are active for all θ. Furthermore, from (5.13) the value for ψ is obtained as $\psi(d,\theta) = 1/3(\theta_1 + \theta_2 - d)$, indicating that $\psi(d,\theta)$ is indeed monotonic in θ, and that maximization of ψ results in the critical point given by $\theta^4 = [2,2]$. Again, the signs of the gradients $\partial\psi/\partial\theta_1$ and $\partial\psi/\partial\theta_2$ are both positive, whereas there is no constraint in (5.6) that satisfies this condition. Therefore, in general there is no way of relating the monotonicity of ψ with the monotonicity of the individual constraint functions f_j. From this it should also be clear why the maximization of individual constraint functions does not predict the right critical parameter values. It is also important to note here, that in general the set of active constraints would depend on the

design d, from which it follows that the monotonicity of $\psi(d,\theta)$ may very well depend on the design d, when $m > n_z + 1$. For instance, in the set of constraints (4.16), ψ is not monotonic in θ for $d = 1$, as shown in Figure 4.3c. However, as can be seen from Figure 5.5, for $d = 0.5$, ψ is monotonic in θ.

Unfortunately, the analysis presented above would seem to be too restrictive to be applicable for the general case of nonlinear constraint functions. However, the analysis does provide some insight on the condition for monotonicity of $\psi(d,\theta)$, which might suggest that it would be worth to investigate further this point.

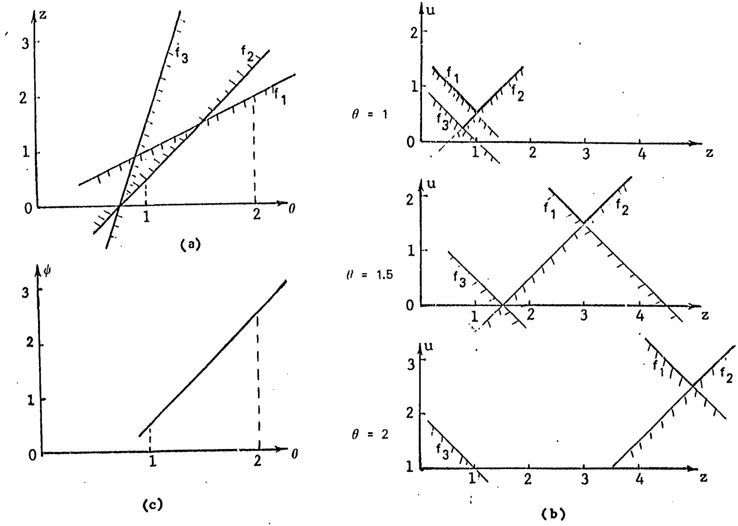

Figure 5.5: Feasible region and $\psi(d,\theta)$ for the constraints (4.16) with d = 0.5

5.5 NOMENCLATURE FOR THE NUMERICAL EXAMPLE PROBLEMS

A	:	Heat transfer area of the heat exchanger, m^2
C	:	Annual cost, \$/yr
C_{Ao}	:	Concentration of reactant in the feed stream, kgmole/m^3
C_{A1}	:	Concentration of reactant in the product stream, kgmole/m^3
C_p	:	Heat capacity of reaction mixture, kJ/kgmole K
(E/R)	:	Ratio of activation energy to gas constant, K
F_o	:	Feed flowrate, kgmole/hr
F_1	:	Flowrate of recycle, kgmole/hr
F_s	:	Flowrate of steam, kg/hr
F_w	:	Flowrate of cooling water, kg/hr
FC	:	Heat capacity flowrate, kJ/hr K
ΔH_{rxn}	:	Heat of reaction, kJ/kgmole
k_R	:	Reaction rate constant, hr^{-1}
N	:	Total number of vertices considered for design, (integer)
T_o	:	Temperature of feed stream, K
T_1	:	Reaction temperature, K
T_2	:	Recycle temperature, K
T_i	:	Temperature of stream i, K
T_w	:	Temperature of cooling water, K (subscript 1 - inlet, subscript 2 - outlet)
U, U^i	:	Overall heat transfer coefficient, kJ/m^2 hr K
$\hat{V}$	:	Reactor volume (design capacity), m^3
V	:	Reaction volume, m^3
ω^i	:	weighting factor corresponding to vertex i.

Superscript i refers to the vertex i (period of operation), i=1,2,....N

CHAPTER 6

CONCLUSIONS AND RECOMMENDATION FOR FURTHER RESEARCH

The problem of optimal design of flexible chemical plants has been studied in this thesis. In order to account for changes in operating conditions and yet meet the design specifications, systematic procedures have been proposed to introduce the required flexibility in the optimal design of a chemical plant. It is shown that there can be two classes of problems in optimal design of flexible chemical plants - the deterministic multiperiod problem and the problem of design under uncertainty. The problem formulation and solution strategies for both these classes of problems have been discussed.

6.1 DETERMINISTIC MULTIPERIOD PROBLEM

If the operating conditions are given as a discrete sequence, the problem of optimal design can be formulated by using the *deterministic multiperiod* model. This leads to a nonlinear program with *block-diagonal structure* in the constraints, wherein the main computational difficulty arises because of a large number of decision variables involved. It is shown that existing decomposition techniques cannot be applied effectively to circumvent this problem. A new *decomposition scheme* has been proposed here, based on a *projection-restriction strategy* which exploits very effectively the mathematical structure of the problem and the fact that many inequality constraints become active at an optimum design solution. The performance of this projection-restriction strategy has been analyzed to show that significant gains in computational effort can be achieved with this technique. The relationship between the performance of the projection-restriction strategy and the size (number of periods) of the design problem being solved is discussed, wherein the effect of having many active constraints at the solution is clearly shown. The results of the numerical example show a linear increase in computational time with the number of periods. It is interesting to note that the proposed decomposition scheme does not presuppose the use of any particular type of optimization algorithm for solving the nonlinear programming problem, and is conceptually quite general in its application for the type of design problems discussed.

As is shown, the efficacy of the proposed decomposition technique is dependent on the number of inequality constraints that actually become active at the optimum design solution. It should be noted that this is not a serious limitation of the technique, since it is a common observation in design that many inequality constraints do in fact become active at the optimum solution of a design problem. On the other hand, the decomposition technique exploits this additional feature of the design problem, thus resulting in a very efficient solution strategy.

It is mentioned that there are two important questions to be considered for a successful implementation of the proposed projection-restriction strategy. One of them is the problem of *finding an initial feasible point* for the problem. The proposed method for this purpose consists in replacing the objective function in the original design problem by the sum of squares of the constraint violations, and solving it by optimizing alternatively with respect to the design and control variables. This is found to be very efficient since it does decompose the problem to a manageable size, and handles an objective function that is relatively well behaved.

The second question is concerned with a *procedure for eliminating the variables* in the restriction step, by using the active constraints, for reducing the number of decision variables of the nonlinear program. This involves the selection of an appropriate set of new control variables so that the resulting system of equations is solvable. Here again, a procedure has been proposed, which is in fact shown to be applicable also in the general context of *selection of decision variables and detection of redundancy in rectangular systems.*

6.2 DESIGN UNDER UNCERTAINTY

When the values of some of the parameters are either known only approximately or they actually vary or fluctuate during the operation, the design problem has to be formulated so as to account for this uncertainty in such parameters. Since it is usually difficult to obtain a prior knowledge of the probability distribution for these parameters, before the implementation of a design, it is assumed that the range of values for these parameters is available. However, in cases where the probability distribution is known, the bounds on these parameter values may be obtained as appropriate confidence limits for these values which can be determined from such data.

Having fixed the range of values for these parameters, which are

assumed to be independent, the objective is to propose a design strategy that can guarantee feasible steady state operations of the plant for every such value of these parameters. That is, the plant is designed with the required flexibility, so that depending on the actual values of these parameters (within the given bounds} the operation of the plant can be suitably adjusted to meet the design specifications. A rigorous mathematical formulation which represents this design strategy has been shown to be a *two-stage nonlinear infinite program*, wherein optimization is to be performed on the set of design and control variables so as to satisfy the constraints for every possible value of the parameters within the specified bounded polyhedral region, while minimizing the expected value of the annual cost function. To guarantee feasibility of the design for every possible realization of the parameters, a logical constraint is incorporated in the problem. This *feasibility constraint* ensures that for every allowable value of the parameters that may be encountered during the operation, appropriate values of control variables can be chosen so as to satisfy the design constraints.

The main computational difficulty with the above formulation arises because of the infinite number of variables and an infinite number of constraints involved. By selecting a discrete set of parameter values and approximating the expected value of the cost function by a weighted average, the number of control variables can be reduced to be finite. To circumvent the problem of an infinite number of constraints, an equivalence for the feasibility constraint has been established, which leads to a *max-min-max* constraint. This max-min-max constraint has also proved to be very useful in obtaining a deeper understanding of the feasibility aspect of the design problem. The following are the main results obtained from an analysis of the properties of the max-min-max constraint in relation to the design problem.

- It is shown that for a given design and a fixed parameter value, the *max-min-max* problem provides a *measure of (in)feasibility of the operation.*

- A critical parameter value corresponds to the smallest degree of feasibility for operation, or the largest degree of infeasibility.
- In general there can be more than a single critical parameter value that must be considered for design, in order to ensure feasibility.
- If the constraints are *convex* the critical points must lie at the *vertices* of the polyhedral region of parameters.

Based on these results an algorithm has been proposed which transforms the problem of design under uncertainty into an iterative multiperiod design problem, wherein each period would correspond to operating the plant with a distinct value of the parameters. The proposed algorithm leads to an efficient solution procedure, as has been shown by two example problems.

6.3 RECOMMENDATIONS FOR FURTHER RESEARCH

In the formulation of the *deterministic multiperiod design problem* the constraints have been considered to be coupled in the design variables, but uncoupled in the state and control variables among the different periods. If in fact there is a subset of constraints interconnecting these variables in different periods, the problem will then have a bordered block-diagonal structure in the constraints. However, the proposed projection-restriction strategy could still be applicable if in the linking constraints only a few coupled state and control variables occur, as then these can be treated as design variables to recover the block-diagonal structure. This problem of coupling constraints needs further work so as to devise a scheme which can render the projection-restriction strategy applicable, while still maintaining its efficiency as a promising solution procedure. Additional work is required also for a theoretical investigation of the convergence properties of the proposed projection-restriction strategy. It should be noted here, that Ritter (1973) has

presented a proof for his version of the decomposition technique when applied to linear constraints. Regarding further numerical experience with the projection-restriction strategy, a general computer program incorporating this strategy has been the objective of a Master's thesis project (Avidan, 1982). Such a capability will certainly be useful for further investigations into the possible refinements of the proposed strategy so as to make it applicable also for the more general case of multiperiod design problems with coupling constraints.

In the *problem of optimal design with uncertain parameter values*, one of the factors in the proposed solution Algorithm 2 that requires further investigation is the question of avoiding the cumulative build-up of the number of parameters to be considered for the multiperiod design problem in Step 2 of Algorithm 2. It is suggested that this could be achieved by removing at every iteration of Algorithm 2, at least one parameter point, on the basis of the minimum value of $\psi(d,\theta)$, which indicates a parameter value with a feasible region having the largest depth and hence most likely to remain feasible for small changes in design. This procedure would seem to be a very good heuristic, in the absence of a more efficient and theoretically valid approach which can still guarantee feasibility for the entire range of parameter values. Further investigation is needed in this direction, to study the scope and limitations of this approach.

As for Step 3 of Algorithm 2, it would be desirable to devise a more efficient procedure for solving the max-min-max problem when a large number of parameters is involved. Note that in the proposed algorithm the value of $\psi(d,\theta)$ has to be determined explicitly for every vertex point in the given polyhedral region of the parameter space, in order to find the vertex that maximizes $\psi(d,\theta)$. Therefore, this approach for locating the critical points could become quite inefficient as the number of uncertain parameters increases, since the corresponding number of vertices increases exponentially.

One possible solution strategy would be to linearize the constraints and perform a parametric analysis of the resulting linear program to determine the critical parameter points, along with an appropriate procedure for bounding the value of ψ to account for the error in the above linearization. However, to make this a valid procedure for finding critical points in the nonlinear case would seem to be a nontrivial task which would require considerable research work.

Note that the uncertain parameters are considered to be varying independently of one another, both in the mathematical formulation of the design problem in Chapter 4, as well as in presenting the solution algorithms in Chapter 5. It should be pointed out here, that the two-stage programming formulation given in Chapter 4 is in fact quite general, and is valid even when the parameter variations are linearly constrained (through their interdependence). Also, the basic idea of the proposed solution algorithm can indeed be applicable for the case of interdependent parameter variations, if a suitable procedure can be developed for locating the critical points in Step 3 of Algorithm 2. However, this would involve the ability to identify the extreme points of the constrained convex region of parameters, and in general it would be desirable to locate the critical points through a complete solution for the max-min-max problem.

It is also important to note that the assumption of convexity of the constraint functions is only a sufficient condition for the location of critical points to be at the vertices (extreme points) of the given polyhedral (convex) region in the parameter space. In fact, a weaker condition would be to relax the assumption of convexity of the constraint functions, and instead impose the condition of convexity of the feasible region. Also, since in the nonconvex case the critical points may not correspond to vertices, it would be desirable to devise a procedure for solving explicitly the max-min-max constraint. However, the complexity of this constraint, and the fact that it can

lead to a nondifferentiable problem with multiple local solutions would appear to make it very difficult to have an efficient solution procedure valid for the general case.

6.4 CONCLUSIONS

This research work has resulted in the following contributions to the area of Computer-Aided Chemical Process Design:

* A systematic approach for introducing flexibility in optimal design of chemical plants, that is more rational than other existing approaches.

- A very efficient decomposition scheme based on a projection-restriction strategy for solving multiperiod design problems, which results in significant gains in computational effort.
- A rigorous mathematical formulation for the problem of optimal design with uncertainty in parameter values, for which its interpretation and properties have been investigated.
- An efficient strategy for solving the problem of design under uncertainty that results in designs which can be guaranteed to be both feasible and optimal.
- A procedure for detection of redundancy and selection of decision variables in process design computations, so as to avoid the problems of singularity, in solving systems of equations using tearing procedure.

APPENDIX

SELECTION OF DECISION AND TORN VARIABLES IN PROCESS DESIGN COMPUTATIONS

One of the main requirements of chemical process computations is the solution of sparse systems of nonlinear algebraic equations. For instance, it has been mentioned in Chapter 2 that the state variables are determined by solving the set of equations representing the chemical process system, which are in general nonlinear. Also, in Step 3 of the projection-restriction strategy, the active constraints are added to the set of equality constraints so as to define a new system of equations, for which a selection of decision variables (control variables) has to be obtained. In this case, it is in fact possible that there are more equations than there are variables, and therefore it is desirable that the procedure being developed can also handle such situations. However, very often these systems have several degrees of freedom since the number of variables is usually larger than the number of *independent* equations. In order to derive a solution procedure for these rectangular systems, it is necessary to *identify redundant equations* in order to determine the number of degrees of freedom of the system, and to select a suitable set of decision variables, so that the resulting square system of

equations is solvable. This Appendix presents such a procedure which is applicable not only to the projection-restriction strategy, but also to general design computations.

A.1 INTRODUCTION

In formulating a solution procedure for systems of algebraic equations that arise in process design computations, the set of variables is partitioned into a vector of decision variables and a vector of state variables. The state variables are calculated by solving the system of equations for any fixed values of decision variables. These values are chosen either by the engineer or through an optimization of some appropriate objective function.

It should be noted that the determination of the number of degrees of freedom requires the identification of a set of consistent and non-redundant equations. This in general is a non-trivial problem since the number of equations for a chemical process system can be rather large. Therefore, it is important that automated solution procedures perform this task so as to allow the engineer to *specify all the valid and relevant equations, even if some of these equations are redundant.*

A common approach for deriving a solution procedure for these rectangular systems of equations consists in ordering the equations, selecting decision variables and tearing state variables, so that the occurrence matrix of the system has a *bordered lower triangular* form, as shown in Figure A.1. In solving the system of equations for any fixed values of decision variables, the torn variables are iterated while the remaining state variables are calculated by solving the equations in the lower-triangular portion. Clearly, it is desirable to select an ordering of equations so as to minimize the number of torn variables. A number of algorithms based on this approach have been reported in the literature. See, for example, Lee et al., (1966); Lee and Rudd, (1966);

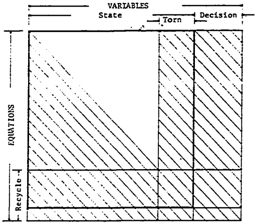

Figure A.1: Bordered lower-triangular structure of the system (A.2)

Christensen and Rudd, (1969); Christensen, (1970); Westerberg and Edie, (1971); Leigh, (1973); Stadtherr et al., (1974); Hernandez and Sargent, (1979). When the system has redundant equations, or if there are singularities associated with certain choices of decision variables, all these algorithms can fail to obtain a square system which is non-singular. In fact, only Edie and Westerberg (1971) present an algorithm for choosing the decision variables to avoid singularity of the resulting square system, but they assume that the original system does not contain redundant equations. The method described below is suitable for detecting redundancy in a given set of equations and selecting an appropriate set of decision variables, so that the resulting system is solvable, without the problem of inconsistency and singularity.

A.2 CONSISTENCY AND SINGULARITY

For the purpose of the present discussion, let the system of nonlinear algebraic equations be

$$h(x) = 0, \quad x \in X \subset R^n, \quad f \subset R^m \tag{A.1}$$

In order to analyze the system of equations it is useful to define the three properties: *structural (in)consistency, algebraic (non-)singularity and numerical (non-)singularity*. The system is said to be *structurally consistent* if by permutations of the rows and columns of its occurrence matrix it is possible to arrive at a rearrangement of the occurrence matrix having non-zero entries along the diagonal. In other words, the system is structurally consistent if and only if it has at least one admissible output set, as shown by Steward (1962). Structural inconsistency arises because of the presence of a subsystem having more equations than the number of state variables defining the subsystem, and therefore always results in a jacobian matrix that is rank deficient.

While *(in)consistency* is a structural property of a system, (non-)singularity is associated with the algebraic and numerical characteristics of a system. The system is *algebraically singular* if and only if its jacobian matrix is rank deficient at all points in the domain X of the system. If the jacobian matrix is rank deficient only at certain points being analyzed, then the system is said to be *numerically singular* at those points. Note that algebraic singularity does not necessarily imply structural inconsistency. An example of this is a consistent system of equations which are linearly dependent. Also note that algebraic non-singularity is indicated by the presence of at least one point where the system is numerically non-singular. Thus, for any system of algebraic equations the following implications hold:

structurally inconsistent $\Rightarrow$ *algebraically singular*

$\Rightarrow$ *numerically singular at every point in the domain* **X**.

and,

numerically non-singular at every point in the domain **X**

$\Rightarrow$ *algebraically non-singular* $\Rightarrow$ *structurally consistent.*

From these implications, it is clear that structural consistency is only necessary but not sufficient to ensure algebraic non-singularity. For successful numerical solution, most algorithms (e.g. Quasi-Newton methods) require that the system of equations be at least algebraically non-singular in the state variables in the region of computation. Algebraic singularity may be caused by structural inconsistency or by the presence of redundant equations or even by a bad choice of decision variables. Thus structural analysis alone is not sufficient for deriving a procedure to eliminate redundant equations and to select a right set of decision variables. As shown below, it is possible to modify existing equation ordering algorithms so as to ensure algebraic non-singularity in the system of equations.

A.3 IDENTIFYING REDUNDANCY AND SELECTING DECISION VARIABLES

First, the rows and columns of the occurrence matrix of the equations in (A.1) are permuted using any of the above cited algorithms, so as to obtain a bordered lower triangular form. The reordered system of equations $f'(x') = 0$ can be represented as shown in Figure A.2, by the two sets of equations:

$$\begin{aligned} s(u,v) &= 0 \\ r(u,v) &= 0 \end{aligned} \tag{A.2}$$

where $[f']^T = [s^T, r^T]$ and $[x']^T = [u^T, v^T]$.

Here, s is the set of non-recycle equations with lower triangular structure in the state variables v, which can therefore be solved sequentially for the vector v given a value of u, and r corresponds to the set of recycle equations. Since v can be treated as an implicit function of u, the system of equations in (A.2) can be reduced to the form:

$$r(u,v(u)) = 0 \tag{A.3}$$

The vector u includes the set of decision variables d as well as the vector of torn variables t, that is, $u^T = (d^T, t^T)$. For any chosen value of d, the system may be solved by iterating on the torn variables t, to reduce the residuals of the equations in (A.3) to zero. The partitioning of the vector u is in general not trivial, and requires careful consideration to avoid singularities, as shown below.

The subsystem s in (A.2) can be assumed to be algebraically non-singular because of its lower-triangular structure. Hence, the algebraic singularity of the system can be analyzed through the jacobian $J_c(r,u)$ of (A.3), which is in fact the *constrained jacobian* of the original system (A.2) corresponding to the recycle equations. This matrix is given by:

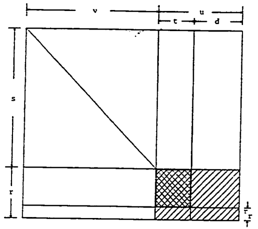

Figure A.2: Constrained jacobian matrix for the system (A.2)

$$J_c(r,u) = \frac{\partial r}{\partial u} - \frac{\partial r}{\partial v}\left(\frac{\partial s}{\partial v}\right)^{-1}\frac{\partial s}{\partial u} \tag{A.4}$$

and is indicated in Figure A.2 by the entire shaded area. In order to analyze the singularity of the system in (A.3), we need to determine the rank of this jacobian matrix $J_c(r,u)$. With this, a corresponding non-singular submatrix J_c^* of highest rank can be found as shown in Figure A.2 by the darker area. The variables in u that form the columns of this submatrix J_c^* will then be chosen as the torn variables t. The remaining variables in u will correspond to the set of decision variables d, and its dimensionality determines the number of degrees of freedom of the system. Those equations r_r that are not included in the rows of this submatrix J_c^* will be deleted, since they are either redundant or inconsistent. Thus it can be ensured that the jacobian matrix of the resulting square system is of full rank and hence non-singular (Halemane and Grossmann, 1981). Since there can be several possible choices for the submatrix J_c^*, it is also possible to incorporate any preferred choice for the decision variables in the above procedure.

In practice, the jacobian matrix $J_c(r,u)$ given in (A.4) can be evaluated numerically at the current point by performing small perturbations in the vector u. Since equation ordering algorithms tend to minimize the dimension of the vector u, the work involved in this procedure is often very moderate. The rank of the jacobian matrix $J_c(r,u)$ and a corresponding non-singular square submatrix J_c^* of highest rank can be obtained by performing a Gaussian elimination on this matrix. For this elimination, the procedure presented by Westerberg et al. (1979) can be applied. Also, in order to ensure that the redundant equations are always deleted and the remaining system is non-singular, the specified tolerances for the pivots should be larger than the maximum round-off error that may be accumulated in the elimination process.

It is to be noted here that by analyzing the jacobian (A.4) at a single point, there is sometimes the danger of not being able to distinguish between a non-singular jacobian matrix that is ill-conditioned and a truly singular matrix. This can cause the deletion of a non-redundant equation and introduce more design variables than required. To guard against this possibility, one may evaluate the jacobian at different points. However, a better strategy would be the following procedure:

Whenever a numerical solution is obtained for the square system, the residuals of the deleted equations are evaluated to check whether they are satisfied within a given tolerance. If any of these deleted equations is not satisfied, it indicates that the number of degrees of freedom of the system has been over-estimated, and therefore requires some decision variables to be converted to state variables. This can be done by a reanalysis of the constrained jacobian at the current point. It should be noted that one can expect in general to perform only one analysis of the constrained jacobian, except for severely ill-conditioned problems.

A.4 EXAMPLE

To illustrate the ideas presented above, the example of a binary isothermal flash system (Edie and Westerberg, 1971) is considered, as shown in Figure A.3. This can be represented by the set of equations (a) to (l) as:

$$P_1^o - P_1^o(T) = 0 \quad \text{(a)}$$

$$P_2^o - P_2^o(T) = 0 \quad \text{(b)}$$

$$P_1^o - K_1 P = 0 \quad \text{(c)}$$

$$P_2^o - K_2 P = 0 \quad \text{(d)}$$

$$y_1 - K_1 x_1 = 0 \quad \text{(e)}$$

$$y_2 - K_2x_2 = 0 \tag{f}$$

$$z_1 + z_2 - 1 = 0 \tag{g}$$

$$x_1 + x_2 - 1 = 0 \tag{h}$$

$$y_1 + y_2 - 1 = 0 \tag{i}$$

$$x_1L + y_1V - z_1F = 0 \tag{j}$$

$$x_2L + y_2V - z_2F = 0 \tag{k}$$

$$L + V - F = 0 \tag{l}$$

In this case all the equations are clearly valid, but they contain one redundant equation, for instance, equation (l). However, this equation will be included in the system so as to illustrate the proposed procedure presented earlier.

Table A.1 shows a structure obtained by equation ordering, with the entries corresponding to the non-zero elements of the jacobian matrix of the original system. The constrained jacobian $J_o(r,u)$ corresponding to the recycle equations, as given by the expression in (A.4) is obtained algebraically and shown in Table A.2. It is clear that the rank of this matrix is *two*, as it can be seen in Table A.2 that the row (k) is obtained by adding row (l) to L times row (h). Hence, one of these three recycle equations is redundant and needs to be deleted. Table A.3 gives the possible choices for decision variables, that arise by the deletion of any one of these three equations. Further analysis as for a final selection of a set of decision variables can be made depending upon the requirements of the particular problem at hand.

A.5 DISCUSSION

The above example shows how the proposed method can be used to identify redundant equations and to select a proper set of decision variables, so as to result in a non-singular square system which can be solved numerically using suitable equation solving algorithms. Also, it is important to note that, by this method one is able to determine the right number of degrees of freedom of a system, which is essential in both the analysis and design of any process system. In using the proposed method, the engineer is allowed to specify all relevant and valid equations that describe the system, even if some of the equations are redundant. The analysis required to identify redundancy and to select a right set of decision variables as presented in the proposed method, can be incorporated in an automated solution procedure.

The numerical information on the jacobian matrix, which is required for the analysis presented earlier, can in fact be used also as an initial estimate in the updating formula of a Quasi-Newton method (e.g. see Broyden, 1965) when solving the recycle equations. The proposed method in itself, is not limited by the numerical evaluation and analysis of the jacobian matrix at any given point. This is clear from the example wherein no numerical evaluation has been used. If an automatic algebraic manipulating package is assumed to be available, it should be possible to detect the algebraic singularity of the jacobian matrix, and hence of the system. It should be noted here that although this procedure seems to be directly applicable only when using a tearing procedure for equation solving, there is no reason why it cannot be extended for other types of equation solving algorithms.

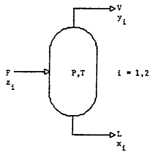

Figure A.3: Binary isothermal flash system

Table A.1: Structure of the binary isothermal flash system as obtained by equation ordering

	P_1^o	P_2^o	K_1	K_2	y_2	x_1	x_2	z_2	L	V	F	z_1	y_1	P	T
(a)	1														a
(b)		1													b
(c)	1		$-P$											$-K_1$	
(d)		1		$-P$										$-K_2$	
(i)					1								1		
(e)			$-x_1$			$-K_1$							1		
(f)				$-x_2$	1		$-K_2$								
(g)								1				1			
(j)						L			x_1	y_1	$-z_1$	$-F$	V		
(h)						1	1								
(k)					V		L	$-F$	x_2	y_2	$-z_2$				
(l)									1	1	-1				

$$a = \frac{-\partial P_1^o}{\partial T} \qquad b = \frac{-\partial P_2^o}{\partial T}$$

Table A.2: Constrained Jacobian $J_c(r,u)$
for the structure given in Table A.1

	V	F	z_1	y_1	P	T
(h)	0	0	0	$1/K_1 - 1/K_2$	$1/P$	$ax_1/K_1P + bx_2/K_2P$
(k)	$1 - y_1/x_1$	$-1 + z_1/x_1$	F/x_1	$-(V + L/K_2)/x_1$	0	$-Lax_2/K_1P + Lbx_2/K_2P$
(l)	$1 - y_1/x_1$	$-1 + z_1/x_1$	F/x_1	$-(V + L/K_1)/x_1$	$-L/P$	$- La/K_1P$

Table A.3: Possible choices of decision variables for the ordering in Table A.1.

Equation deleted	Decision Variables	Torn Variables
(h)	V, F, z_1, y_1	P, T
	V, F, z_1, P	y_1, T
	V, F, z_1, P	y_1, P
	V, F, P, y_1	z_1, T
	V, F, T, y_1	z_1, P
	V, F, P, T	z_1, y_1
	V, z_1, y_1, P	F, T
	V, z_1, y_1, T	F, P
	V, z_1, P, T	F, y_1
	F, z_1, y_1, P	V, T
	F, z_1, y_1, T	V, P
	F, z_1, P, T	V, y_1
(k) or (l)	V, F, z_1, y_1	P, T
	V, F, z_1, P	y_1, T
	V, F, z_1, T	y_1, P

REFERENCES

Avidan, A. I.
"A Computer Package for the Design of Multiperiod Flexible Chemical Plants",
M.S. Thesis, Carnegie-Mellon University, Pittsburgh, (1982)

Avriel, M. and D. J. Wilde
"Engineering Design Under Uncertainty",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 8, No. 1, pp 124-131, (1969)

Bandler, J. W.
"Optimization of Design Tolerances Using Nonlinear Programming",
J. Opt. Theory Appl., Vol. 15, No. 1, pp 99-114, (1974)

Bandler, J. W., P. C. Liu and H. Tromp
"A Nonlinear Programming Approach to Optimal Design Centering, Tolerancing and Tuning",
IEEE Trans. on Circuits and Systems, Vol. CAS-23, No. 3, pp 155-165, (1976)

Book, N. L. and F. Ramirez
"The Selection of Design Variables in Systems of Algebraic Equations",
A.I.Ch.E. J., Vol. 22, pp 55-66 ,(1976)

Brosilow, C. and L. Lasdon
"A Two-Level Optimization Technique for Recycle Processes",
A.I.Ch.E.-Inst. Chem. E. Symp. Ser. No. 4, pp 75-83, (1965)

Broyden, C. G.
"A Class of Methods for Solving Nonlinear Simultaneous Equations",
Math. Comp., Vol. 19, pp 577-593, (1965)

Charnes, A. and W. W. Cooper
"Chance Constrained Programming",
Manag. Sc., Vol. 6, pp 73-78, (1959)

Christensen, J. H.
"The Structuring of Process Optimization",
A.I.Ch.E. J., Vol. 16, No. 2, pp 177-184, (1970)

Christensen, J. H. and D. F. Rudd
"Structuring Design Computations",
A.I.Ch.E. J., Vol. 15, No. 1, pp 94-100, (1969)

Danskin J. M.
"The Theory of Max-Min and its applications to weapons allocation problems"
Springer-Verlag, (1967)

Demyanov, V. F. and V. N. Malozemov
"Introduction to MiniMax",
Translated from Russian by D. Louvish,
A Halsted Press Book, John Wiley, N.Y. (1974)

Director, S. W. and G. D. Hachtel
"The Simplicial Approximation Approach to Design Centering",
IEEE Trans. on Circuits and Systems, Vol. CAS-24, No. 7, pp 363-372 (1977)

Dittmar, R. and K. Hartman
"Calculation of Optimal Design Margins for Compensation of Parameter Uncertainty",
Chem. Eng. Sci., Vol. 31, pp 563-568, (1976)

Edie, F. C. and A. W. Westerberg
"Computer-Aided Design, Part 3: Decision Variable Selection to Avoid Hidden Singularities in Resulting Recycle Calculations",
Chem. Eng. J., Vol. 2, pp 114-124, (1971)

Freeman, R. A. and J. L. Gaddy
"Quantitative Overdesign of Chemical Processes",
A.I.Ch.E. J., Vol. 21, No. 3, pp 436-440, (1975)

Friedman, Y. and G. V. Reklaitis
"Flexible Solutions to Linear Programs Under Uncertainty: Inequality Constraints",
A.I.Ch.E. J., Vol. 21, No. 1, pp 77-83, (1975a)

Friedman, Y. and G. V. Reklaitis
"Flexible Solutions to Linear Programs Under Uncertainty: Equality Constraints",
A.I.Ch.E. J., Vol. 21, No. 1, pp 83-90, (1975b)

Geoffrion, A. M.
"Elements of Large-Scale Mathematical Programming",
Manage. Sci., Vol. 16, pp 652-675, (1970)

Grigoriadis, M. D.
"A Projective Method for Structured Nonlinear Programs",
Math. Prog., Vol. 1, pp 321-358, (1971)

Grigoriadis, M. D. and K. Ritter

"A Decomposition Method for Structured Linear and Nonlinear Programs", J. Comp. and Sys. Sc., Vol. 3, pp 335-360, (1969)

Grossmann, I. E.

"Problems in the Optimum Design of Chemical Plants", Ph.D. Thesis, University of London, U.K., (1977)

Grossmann, I. E. and K. P. Halemane

"A Decomposition Strategy for Designing Flexible Chemical Plants", to appear, A.I.Ch.E. J., (1982)

Grossmann, I. E. and R. W. H. Sargent

"Optimum Design of Chemical Plants with Uncertain Parameters", A.I.Ch.E. J., Vol. 24, No. 6, pp 1021-1028, (1978)

Grossmann, I. E. and R. W. H. Sargent

"Optimum Design of Multipurpose Chemical Plants", Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 18, No. 2, pp 343-348, (1979)

Halemane, K. P. and I. E. Grossmann

"A Remark on the paper 'Theoretical and Computational Aspects of the Optimal Design Centering, Tolerancing and Tuning Problem'", IEEE Trans. on Circuits and Systems, Vol. CAS-28, No. 2, pp 163-164, (1981)

Halemane, K. P. and I. E. Grossmann

"Selection of Decision and Torn Variables in Process Design Computations", Proceedings of the 1981 Summer Computer Simulation Conference at Washington, D.C., pp 230-234, (1981)

Halemane, K. P. and I. E. Grossmann
"Optimal Process Design Under Uncertainty",
paper presented at the 74th Annual Meeting of the A.I.Ch.E., New Orleans, (1981)

Han, S. P.
"A Globally Covergent Method for Nonlinear Programming",
J. Opt. Theory Appl., Vol. 22, No. 3, pp 297-309, (1977)

Hernandez, R. and R. W. H. Sargent
"A New Algorithm for Process Flowsheeting",
paper # 6B.2, Comp. Chem. Eng., Vol. 3, pp 363-371, (1979)

Hettich, R. (Ed.)
"Semi-Infinite Programming",
Springer-Verlag, (1979)

Johns, W. R., G. Marketos and D. W. T. Rippin
"The Optimal Design of Chemical Plant to Meet Time-varying Demands in the Presence of Technical and Commercial Uncertainty",
paper presented at the Institution of Chemical Engineers Design Congress, Birmingham, U.K., (1976)

Kilikas, A. C. and H. P. Hutchison
"Process Optimisation Using Linear Models",
Comput. and Chem. Eng., Vol. 4, pp 39-48, (1980)

Kittrel, J. R. and C. C. Watson
"Don't Overdesign Process Equipment",
Chem. Eng. Prog., Vol. 62, No. 4, pp 79-83, (1966)

Knopf, F. C., M. R. Okos and G. V. Reklaitis
"Optimal Design of Batch/Semicontinuous Processes",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 21, No. 1, pp 79-86, (1982)

Lasdon, L. S.
"Optimization Theory for Large Systems",
Collier-MacMillan Ltd., London, (1970)

Lashmet, P. K. and S. Z. Szczepanski
"Efficiency, Uncertainty and Distillation Column Overdesign Factors",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 13, No. 2, pp 103-106, (1974)

Lee, W., J. H. Christensen and D. F. Rudd
"Design Variable Selection to Simplify Process Calculations",
A.I.Ch.E. J., Vol. 12, pp 1104-1110, (1966)

Lee, K. F., A. H. Masso and D. F. Rudd
"Branch and Bound Synthesis of Integrated Process Designs",
Ind. Eng. Chem. (Fund.), Vol. 9, pp 48-58, (1970)

Lee, W. and D. F. Rudd
"On the Ordering of Recycle Calculations",
A.I.Ch.E. J., Vol. 12, pp 1184-1190, (1966)

Leigh, M.
"A Computer Flowsheeting Programme Incorporating Algebraic Analysis of the Problem Structure",
Ph.D. Thesis, University of London, U.K., (1973)

Loonkar, Y. R. and J. D. Robinson
"Minimization of Capital Investment for Batch Processes",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 9, No. 4, pp 625-629, (1970)

Malik, R. K. and R. R. Hughes

"Optimal Design of Flexible Chemical Processes",
paper # 8B.5, Comp. Chem. Eng., Vol. 3, pp 473-485, (1979)

Mangasarian, O. L.

"Nonlinear Programming",
McGraw Hill, (1969)

Nishida, N., A. Ichikawa and E. Tazaki

"Synthesis of Optimal Process Systems with Uncertainty",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 13, No. 3, pp 209-214, (1974)

Oi, K., H. Itoh and I. Muchi

"Improvement of Operational Flexibility of Batch Units by a Design Margin",
paper presented at the 12^{th} Symposium on Computer Applications in Chemical Engineering, Montreux, Switzerland, April 8-11, (1979)

Polak, E. and A. Sangiovanni-Vincentelli

"Theoretical and Computational Aspects of the Optimal Design Centering, Tolerancing and Tuning Problem",
IEEE Trans. on Circuits and Systems, Vol. CAS-26, No. 9, pp 795-813, (1979)

Powell, M. J. D.

"A Fast Algorithm for Nonlinearly Constrained Optimization Calculations",
paper presented at the 1977 Dundee Conference on Numerical Analysis, in *"Numerical Analysis"*, Dundee 1977, eds. A. Dodd and B. Eckman, # 630, Lecture Notes in Math., Springer-Verlag.

Ritter, K.

"A Decomposition Method for Structured Nonlinear Programming Problems",

pp 399-413, in D. Himmelblau (Ed.), *"Decomposition of Large-Scale Problems"*, North- Holland, Amsterdam, (1973)

Rockefellar, R. T.

"Convex Analysis",

Princeton University Press, (1970)

Rosen, J. B. and J. C. Ornea

"Solution of Nonlinear Programming Problems by Partitioning",

Manage. Sci., Vol. 10, pp 160-173, (1963)

Rudd, D. F. and C. C. Watson

"Strategy of Process Engineering",

John-Wiley, New York, (1968)

Sargent, R. W. H. and B. A. Murtagh

"Projection Methods for Nonlinear Programming",

Math. Prog., Vol. 4, pp 245-268, (1973)

Sparrow, R. E., J. F. Graham and D. W. T. Rippin

"The Choice of Equipment Sizes for Multiproduct Batch Plants - Heuristics vs. Branch and Bound",

Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 14, No. 3, pp 197-203, (1975)

Stadtherr, M. A., W. A. Gifford and L. E. Scriven

"Efficient Solution of Sparse Sets of Design Equations",

Chem. Eng. Sci., Vol. 29, pp 1025-1034, (1974)

Stephanopoulos, G. and A. W. Westerberg

"The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization",

J. Opt. Theory Appl., Vol. 15, No. 3, pp 285-309, (1975a)

Stephanopoulos G. and A. W. Westerberg

"Synthesis of Optimal Process Flowsheets by an Infeasible Decomposition Technique in the Presence of Functional Non-convexities",

Can. J. of Chem. Eng., Vol. 53, pp 551-555, (1975b)

Steward, D. V.

"On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations",

S.I.A.M. Rev., Vol. 4, pp 321-342, (1962)

Stoer, J. and C. Witzgall

"Convexity and Optimization in Finite Dimensions",

Springer-Verlag, (1970)

Suhami, I.

"Algorithms for the Scheduling and Design of Chemical Processing Facilities",

Ph. D. Thesis, Northwestern University, Evanston, Illinois, (1981)

Takamatsu, T., I. Hashimoto and S. Shioya

"On the Design Margin for Process System with Parameter Uncertainty",

J. of Chem. Eng. of Japan, Vol. 6, No. 5, pp 453-457, (1973)

Umeda, T., A. Shindo and E. Tazaki

"Optimal Design of Chemical Process by Feasible Decomposition Method",

Ind. Eng. Chem. (Proc. Des. De.), Vol. 11, pp 1-8, (1972)

Watanabe, N., Y. Nishimura and M. Matsubara
"Optimal Design of Chemical Processes Involving Parameter Uncertainty",
Chem. Eng. Sci., Vol. 28, pp 905-913, (1973)

Wen, C. Y. and T. M. Cheng
"Optimal Design of Systems Involving Parameter Uncertainty",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 7, No. 1, pp 49-53, (1968)

Weisman, J. and A. G. Holzman
Optimal Process System Design Under Conditions of Risk",
Ind. Eng. Chem. (Proc. Des. Dev.), Vol. 11, No. 3, pp 386-397, (1972)

Westerberg, A. W. and F. C. Edie
"Computer-Aided Design, Part 2: An Approach to Convergence and Tearing in the Solution of Sparse Equation Sets",
Chem. Eng. J., Vol. 2, pp 17-25, (1971)

Westerberg, A. W., H. P. Hutchison, R. L. Motard and P. Winter
"Process Flowsheeting",
Cambridge University Press, pp 42-45, (1979)

CORRIGENDUM

Page	Line	Printed As	Read As
90	2	Table 5.5a	Table 5.5b
90	3	A=9.2 m^2	A=9.66 m^2
90	3	Cost: 10110 $/yr	Cost: 10902 $/yr